

# Location-Free Spectrum Cartography

Yves Teganya, Daniel Romero, Luis Miguel Lopez-Ramos,  
and Baltasar Beferull-Lozano

**Abstract**—Spectrum cartography constructs maps of metrics such as channel gain or received signal power across a geographic area of interest using spatially distributed sensor measurements. Applications of these maps include network planning, interference coordination, power control, localization, and cognitive radios to name a few. Since existing spectrum cartography techniques require accurate estimates of the sensor locations, their performance is drastically impaired by multipath affecting the positioning pilot signals, as occurs in indoor or dense urban scenarios. To overcome such a limitation, this paper introduces a novel paradigm for spectrum cartography, where estimation of spectral maps relies on features of these positioning signals rather than on location estimates. Specific learning algorithms are built upon this approach and offer a markedly improved estimation performance than existing approaches relying on localization, as demonstrated by simulation studies in indoor scenarios.

*Keywords*—Spectrum cartography, location-free cartography, kernel-based learning, spectrum map.

## B.1 Introduction

Spectrum cartography constructs maps of a certain channel metric, such as received signal power, power spectral density (PSD), or channel gain over a geographical area of interest by relying on measurements collected by radio frequency (RF) sensors [1–3]. The obtained maps are of utmost interest in a number of tasks in wireless communication networks, such as network planning, interference coordination, power control, and dynamic spectrum access [4–6]. For instance, power maps can be useful in network planning since the former indicate areas of weak coverage, thus suggesting locations where new base stations must be deployed. Since PSD maps characterize the distribution of the RF signal power per channel over space, they can play a major role in increasing frequency reuse to mitigate interference. These maps may also be of interest to speed up hand-off in cellular networks since they enable mobile users to determine the power of all channels at a given location without having to spend time measuring it. Additional use cases may include cognitive radios, where secondary users aim at exploiting underutilized spectrum resources in the space-frequency-time domain, or source localization, where the locations of certain transmitters may be estimated by inspecting a map [2].

Existing methods for mapping RF power apply spatial interpolation or regression techniques to power measurements collected by spatially distributed sensors. Some of these methods include kriging [1, 7, 8], orthogonal matching pursuit [3], matrix completion [9], dictionary learning [10, 11], sparse Bayesian learning [12], or kernel-based learning [13, 14]. Since these works can only map power distribution across space but not across frequency, different schemes have been devised to construct PSD maps, for instance by exploiting the sparsity of power distributions over space and frequency with a basis expansion model [2, 15] or by leveraging the framework of kernel-based learning [4]. Rather than mapping power, other families of methods construct channel-gain maps using Kriged Kalman filtering [16], non-parametric regression in reproducing kernel Hilbert spaces (RKHSs) [17], low rank and sparsity [18], or hidden Markov random fields [19].

All the aforementioned schemes require accurate knowledge of the sensor locations. For this reason, they will be collectively referred to as *location-based (LocB) cartography*. However, location is seldom known in practice and therefore must be estimated from features such as the received signal strength, the time (difference) of arrival (T(D)oA), or the direction of arrival (DoA) of positioning pilot signals transmitted by satellites (e.g. in GPS) or terrestrial base stations (e.g. in LTE or WiFi [20]) [21, 22]. Unfortunately, accurate location estimates are often not available in practice due to propagation phenomena affecting those pilot signals such as multipath, which limits the applicability of existing cartography techniques, especially in indoor and dense urban scenarios. To see the intuition behind this observation, Figs. B.1a and B.1b respectively show the  $x$  and  $y$  coordinates of the location estimates obtained by applying a state-of-the-art localization algorithm to TDoA measurements of 5 pilot signals received in free space (details of the specific simulation setting can be found in Sec. B.5). On the other hand, Figs. B.1c and B.1d depict the same estimates but in an indoor propagation scenario. As observed, the estimates in the second case are neither accurate nor smooth across space, which precludes any reasonable estimate of a spectrum map based on them.

To counteract this difficulty, there are three main types of indoor positioning systems [23]: (i) Those based on ultra-wideband (UWB) [24–26], which require a dedicated infrastructure and relatively high costs, e.g. synchronized anchor nodes in the area where the map has to be constructed. Therefore, localization cannot be carried out in an area where such hardware is not present. (ii) Other indoor positioning systems are based on fingerprinting [23, 27, 28], which involves a manual collection and storage of a dataset. This dataset may comprise the measured power of multiple beacons at a set of known locations. Note that this process is time consuming and typically expensive because a human or robot should physically go through several known locations to take measurements. Furthermore, if there are significant changes in the propagation environment, these methods would require the acquisition of a new dataset. (iii) There exist other indoor positioning systems that combine UWB or fingerprinting with ultrasound [29] or RFID [30]. Thus, they inherit the limitations of (i) and (ii) and require furthermore special sensors and/or line-of-sight propagation conditions. To sum up, all existing cartography schemes require accurate location information, which is not available in dense multipath and indoor scenarios when there are no special localization infrastructure or fingerprinting datasets.

The main contribution of this paper is to address this limitation by proposing the framework of *location-free (LocF) cartography*. The key observation is that inaccurate location estimates introduce significant errors in spectrum map estimation. To bypass this limitation, the proposed approach obtains spectrum maps indexed directly by (or as a function of) features of the received pilot signals. Although many algorithms can be devised within this framework, the present paper develops an algorithm based on kernel-based learning for the sake of exposition. This is not only because of the simplicity, flexibility, and good performance of kernel-based estimators, but also because they have well-documented merits in spectrum cartography [4, 15]. Similarly, the discussion focuses on constructing power maps, but the proposed paradigm carries over to other metrics such as PSD. Remarkably, as a byproduct of skipping the localization step, the resulting cartography algorithm is typically computationally less expensive than its LocB counterparts and does not require additional localization infrastructure or the costly creation of fingerprinting datasets. The second main contribution is a design of pilot signal features tailored to multipath environments. The third contribution is a special technique to accommodate scenarios where a sensor can only extract a subset of those features due to low signal-to-noise ratio (SNR). Finally, the proposed LocF cartography scheme is studied through Monte Carlo simulations in realistic propagation environments. As expected, the proposed scheme outperforms LocB cartography in multipath scenarios, but traditional LocB approaches are still preferable when accurate location estimates are available.

The rest of this paper is structured as follows: Sec. B.2 describes the system model, states the problem, and reviews LocB cartography. Sec. B.3 introduces LocF cartography along with the proposed map estimation algorithm, whereas Sec. B.4 deals with feature design. Numerical tests are presented in Sec. B.5, and conclusions in Sec. B.6.

*Notation:* Scalars are denoted by lowercase letters. Bold uppercase (lowercase) letters denote matrices (column vectors),  $\mathbf{I}_N$  is the  $N \times N$  identity matrix and  $\mathbf{1}$  is the vector of all ones of appropriate dimension. The symbol  $j := \sqrt{-1}$  is the imaginary unit,  $(\cdot)^*$  stands for the complex conjugate, while  $*$  denotes convolution. Furthermore, operators  $(\cdot)^\top$  and  $\|\cdot\|_F$  represent transposition and the Frobenius norm, respectively.

## B.2 Problem Formulation and LocB Cartography

This section formulates the general spectrum cartography problem and reviews the basics of LocB cartography.

The goal is to determine the power  $p(\mathbf{x})$  of a certain channel, termed *channel-to-map (C2M)*, at every location  $\mathbf{x} \in \mathcal{X}$  of a geographical region of interest  $\mathcal{X} \subset \mathbb{R}^d$ , with  $d = 2$  or  $3$ . For example, this C2M can be an uplink or downlink channel of a cellular network as well as a radio or TV broadcasting channel. To this end, a collection of sensors gather  $N$  measurements at locations  $\{\mathbf{x}_n\}_{n=1}^N \subset \mathcal{X}$  not necessarily known. The noisy measurement of the power  $p(\mathbf{x}_n)$  at location  $\mathbf{x}_n$  will be represented as  $\tilde{p}_n$ . Since the sensors collect measurements at multiple locations in  $\mathcal{X}$ , the number of measurements may be significantly greater than the number of sensors.

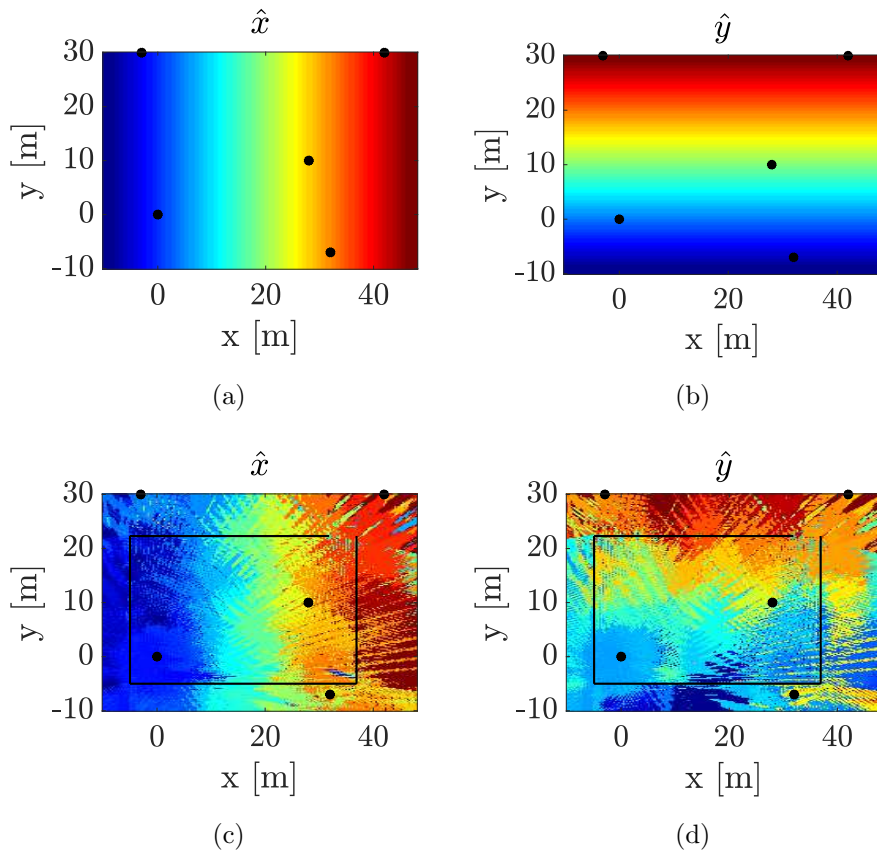


Figure B.1: Estimation of spatial coordinates using TDoA: (a)-(b) in free space, (c)-(d) indoor where the solid black lines represent the walls of the building; the black dots represent the locations of the anchor base stations. The color of each pixel represents the value of the estimated location coordinate at each point in the  $150 \times 150$  grid area. Because location estimates in (a)-(b) coincide with the true locations, they can act as colorbars to the estimates in (c)-(d).

In LocB cartography [1–4, 8–12, 14, 15, 17–19], a fusion center is ideally given pairs  $\{(\mathbf{x}_n, \tilde{p}_n)\}_{n=1}^N$ , which include the exact sensor locations  $\{\mathbf{x}_n\}_{n=1}^N$ , and obtains a function estimate  $\hat{p}(\mathbf{x})$  that provides the power of the C2M at any query location  $\mathbf{x} \in \mathcal{X}$ . With this function, a node at location  $\mathbf{x}$  can determine the power of the C2M if it knows  $\mathbf{x}$ . In practice, however, location is typically unknown and hence the sensor at the  $n$ -th measurement point must estimate  $\mathbf{x}_n$  by relying on pilot signals  $\{y_{l,n}[k]\}_{l=1}^L$ , where  $y_{l,n}[k]$  denotes the  $k$ -th sample of the pilot signal transmitted by the  $l$ -th base station<sup>1</sup> and received at the  $n$ -th measurement point. For convenience, form the  $L \times K$  matrix  $\mathbf{Y}_n$  whose  $(l, k)$ -th entry is  $y_{l,n}[k]$ . Note that these pilot signals are generally transmitted through a separate channel, not necessarily the C2M. However, both channels may coincide, as it occurs in certain cellular communication standards.

<sup>1</sup>Although the discussion assumes for simplicity that the pilot signals are transmitted by terrestrial base stations, the proposed scheme can also be applied when these pilot signals are transmitted by satellites.

From  $\mathbf{Y}_n$ , the sensor at the  $n$ -th measurement point obtains the estimate  $\hat{\mathbf{x}}_n := \hat{\mathbf{x}}(\mathbf{Y}_n)$  of  $\mathbf{x}_n$  by means of some localization algorithm [21, 22]. A fusion center then uses  $\{(\hat{\mathbf{x}}_n, \tilde{p}_n)\}_{n=1}^N$  to obtain an estimate  $\hat{p}(\mathbf{x})$  of the function  $p(\mathbf{x})$ . Therefore, if the location estimates  $\{\hat{\mathbf{x}}_n\}_{n=1}^N$  are noisy, so will be  $\hat{p}(\mathbf{x})$ . If a node at an unknown query location wishes to determine the power of the C2M, it will use the pilot signals  $\mathbf{Y}$  to obtain an estimate  $\hat{\mathbf{x}} := \hat{\mathbf{x}}(\mathbf{Y})$  of its location and will evaluate the map estimate as  $\hat{p}(\hat{\mathbf{x}})$ . In this case,  $\mathbf{Y}$  is a matrix whose  $(l, k)$ -th entry is given by the  $k$ -th sample of the  $l$ -th pilot signal  $y_l[k]$  at the query location. Thus, such an estimation has two sources of error: first, the location estimation error in  $\hat{\mathbf{x}}$  and, second, the map estimation error in  $\hat{p}(\mathbf{x})$ .

**Remark 2** *One may argue that a node can determine the power of the C2M at its location more efficiently by measuring it rather than by receiving the pilot signals, applying a localization algorithm, and evaluating the map. Whereas this may be the case for a single C2M, if the aim is to determine the PSD, the power of many C2Ms, or the impulse response, then the associated measurement time may be prohibitive, which favors the adoption of spectrum cartography approaches.*

## B.3 Location-Free Cartography

This section proposes LocF cartography, which bypasses the localization step involved in all existing cartography approaches. To this end, the LocF cartography problem is formulated as a function estimation task in Sec. B.3.1 and solved via kernel-based learning in Sec. B.3.2.

### B.3.1 Map Estimate as a Function Composition

As detailed in the previous section, existing spectrum cartography techniques are heavily impaired by localization errors since the maps they construct are functions of noisy location estimates. The main idea of the proposed framework is to bypass such a dependence.

To this end, it is worth interpreting LocB cartography from a more abstract perspective. As detailed in Sec. B.2, the LocB map estimate is of the form  $\hat{p}(\hat{\mathbf{x}})$  with  $\hat{\mathbf{x}} := \hat{\mathbf{x}}(\mathbf{Y})$  denoting the output of the selected localization algorithm when the pilot signals are given by  $\mathbf{Y} \in \mathcal{Y}$ . Thus, this estimate can be seen as a function of  $\mathbf{Y}$ , i.e.  $\hat{p}_{\mathbf{Y}}(\mathbf{Y}) := \hat{p}(\hat{\mathbf{x}}(\mathbf{Y}))$ , which can be expressed schematically as:

$$\begin{array}{ccccc} \mathcal{Y} & \xrightarrow{\hat{\mathbf{x}}} & \mathcal{X} & \xrightarrow{\hat{p}} & \mathbb{R} \\ \mathbf{Y} & \longrightarrow & \hat{\mathbf{x}}(\mathbf{Y}) & \longrightarrow & \hat{p}(\hat{\mathbf{x}}(\mathbf{Y})). \end{array} \quad (\text{B.1})$$

As mentioned in Sec. B.2, existing (LocB) cartography approaches obtain an estimate  $\hat{p}$  of  $p$  using the data  $\{(\hat{\mathbf{x}}(\mathbf{Y}_n), \tilde{p}_n)\}_{n=1}^N$  for instance by searching for a function in an RKHS [4, 13, 14]. When  $\hat{\mathbf{x}}(\mathbf{Y})$  is a reasonable estimate of the location  $\mathbf{x}$  at which  $\mathbf{Y}$  has been observed, such a LocB approach works well. However, due to multipath propagation effects impacting the pilot signals in  $\mathbf{Y}$ ,  $\hat{\mathbf{x}}(\mathbf{Y})$  may be very different from  $\mathbf{x}$ , which drastically hinders the estimation of  $p$ . Thus, in those cases where the location

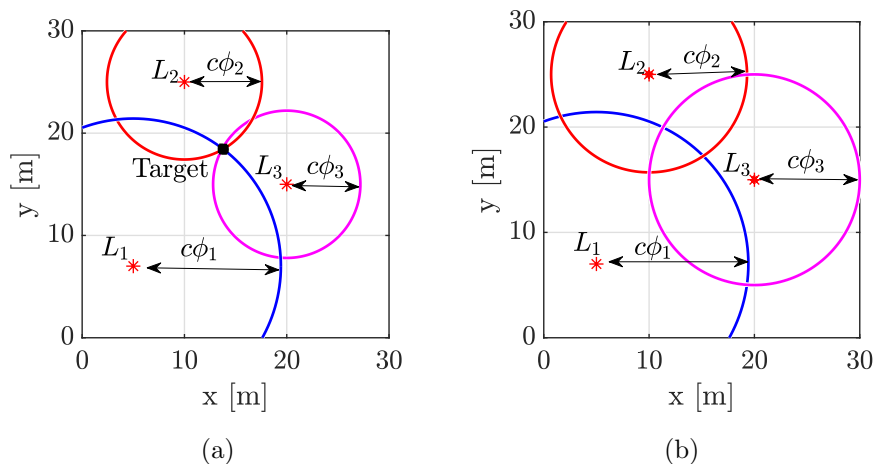


Figure B.2: Multi-lateration using ToA measurements with circles as possible sensor locations: (a) consistent ToA with the sought sensor location being the intersection of the circles (black square) and (b) inconsistent ToA measurements. The red stars represent the locations of the anchor base stations.

estimates  $\{\hat{\mathbf{x}}(\mathbf{Y}_n)\}_{n=1}^N$  are noisy, the resulting estimate  $\hat{p}$ , and consequently  $\hat{p}_{\mathbf{Y}}$ , will be correspondingly noisy.

Since the source of such an error is the dependency of  $\hat{p}_{\mathbf{Y}}(\mathbf{Y}) = \hat{p}(\hat{\mathbf{x}}(\mathbf{Y}))$  on the estimated location  $\hat{\mathbf{x}}(\mathbf{Y})$ , one could think of bypassing this dependence by directly estimating  $\hat{p}_{\mathbf{Y}}$  as a general function of  $\mathbf{Y}$ :

$$\begin{aligned} \mathcal{Y} &\xrightarrow{\hat{p}_{\mathbf{Y}}} \mathbb{R} \\ \mathbf{Y} &\longrightarrow \hat{p}_{\mathbf{Y}}(\mathbf{Y}). \end{aligned} \tag{B.2}$$

When pursuing an estimate of this general form,  $\hat{p}_{\mathbf{Y}}(\mathbf{Y})$  would not be confined to depend on  $\mathbf{Y}$  only through the estimated location. However, finding such an estimate given  $\{(\mathbf{Y}_n, \tilde{p}_n)\}_{n=1}^N$  by searching over a generic class of functions such as an RKHS would be extremely challenging due the so-called *curse of dimensionality* [31, 32]. To intuitively understand this phenomenon, note that the number of input variables of function  $\hat{p}_{\mathbf{Y}}(\mathbf{Y})$  is  $LK$ , typically in the order of hundreds or thousands. Since learning a multivariate function up to a reasonable accuracy generally requires that the number of data points be several orders of magnitude larger than the number of input variables, this approach would need  $N$  to be significantly larger than  $LK$ , and therefore prohibitively large.

To summarize, the structure imposed by (B.2) is too generic, whereas the one imposed by (B.1) is too restrictive. To attain a sweet spot in this trade-off, it is worth decomposing  $\hat{\mathbf{x}}(\mathbf{Y})$  as detailed next. Recall that  $\hat{\mathbf{x}}(\mathbf{Y})$  is the result of applying a localization algorithm to the pilot signals  $\mathbf{Y}$ . For most existing algorithms,  $\hat{\mathbf{x}}(\mathbf{Y})$  can be thought of as the composition of two functions: a function  $\phi : \mathcal{Y} \rightarrow \mathcal{F} \subset \mathbb{R}^M$  that obtains  $M$  features from  $\mathbf{Y}$ , such as T(D)oA or DoA, and a function  $\hat{l} : \mathcal{F} \rightarrow \mathcal{X}$ , that provides a location estimate

$\hat{\mathbf{l}}(\phi)$  given a feature vector  $\phi \in \mathcal{F}$ . In this case,  $\hat{p}_{\mathbf{Y}}(\mathbf{Y})$  can be decomposed as:

$$\begin{aligned} \mathcal{Y} &\xrightarrow{\phi} \mathcal{F} && \xrightarrow{\hat{\mathbf{l}}} \mathcal{X} && \xrightarrow{\hat{p}} \mathbb{R} \\ \mathbf{Y} &\longrightarrow \phi(\mathbf{Y}) && \longrightarrow \hat{\mathbf{l}}(\phi(\mathbf{Y})) && \longrightarrow \hat{p}(\hat{\mathbf{l}}(\phi(\mathbf{Y}))). \end{aligned} \quad (\text{B.3})$$

Observe that the reason why the location estimate  $\hat{\mathbf{x}}(\mathbf{Y}) = \hat{\mathbf{l}}(\phi(\mathbf{Y}))$  is inaccurate in multipath environments is because the algorithm that evaluates  $\hat{\mathbf{l}}$  adopts a model where there is a certain “agreement” among features  $\phi(\mathbf{Y})$ . To see this, consider Fig. B.2, which illustrates the task of estimating the location of a sensor in an area with  $L = 3$  base stations. The features in  $\phi \in \mathbb{R}^M$ , with  $M = L = 3$ , used in this example are noiseless ToA features. For each pilot signal, there is a circle centered at the base station and whose radius equals  $c$  times the ToA, where  $c$  is the speed of light. Thus, when there is no multipath, the ToA features are accurate and the sensor to be located must lie in the intersection of the three circles, as shown in Fig. B.2a. Thus, the localization algorithm (embodied in  $\hat{\mathbf{l}}$ ) just needs to return the location at which these circles intersect. However, in multipath environments, the ToA features obtained from  $\mathbf{Y}$  do not generally equal the time it takes for an electromagnetic wave to propagate from the corresponding base station to the sensor. As a result, the aforementioned circles will not generally intersect; see Fig. B.2b. In other words, the expected agreement among features is absent and, hence, the localization algorithm will return an inaccurate estimate of the position.

In view of these arguments, the key idea in this paper is to pursue estimates  $\hat{p}_{\mathbf{Y}}(\mathbf{Y})$  of the form:

$$\begin{aligned} \mathcal{Y} &\xrightarrow{\phi} \mathcal{F} && \xrightarrow{\hat{d}} \mathbb{R} \\ \mathbf{Y} &\longrightarrow \phi(\mathbf{Y}) && \longrightarrow \hat{d}(\phi(\mathbf{Y})). \end{aligned} \quad (\text{B.4})$$

In this setting, the problem is find an estimate  $\hat{d}(\phi)$  given  $\{(\phi_n, \tilde{p}_n)\}_{n=1}^N$ , where  $\phi_n := \phi(\mathbf{Y}_n)$ . By following this approach, the estimated map  $\hat{p}(\mathbf{Y}) = \hat{d}(\phi(\mathbf{Y}))$  does not involve a high number of inputs as in (B.2) and does not depend on the location estimate as in (B.1). For the latter reason, this approach will be referred to as *LocF cartography*. Since this approach does not need the agreement among entries of  $\phi(\mathbf{Y})$  illustrated in Fig. B.2b, it is expected to outperform traditional spectrum cartography methods when such an agreement is not present, as occurs in multipath environments.

### B.3.2 Kernel-based Power Map Learning

This section applies kernel-based learning to provide an algorithm capable of learning the function  $\hat{d}$  introduced in Sec. B.3.1.

Given pairs  $\{(\phi_n, \tilde{p}_n)\}_{n=1}^N$ , where  $\phi_n := \phi(\mathbf{Y}_n)$ , the problem can be informally stated as finding a function  $\hat{d}$  that satisfies two conditions: CO1)  $\hat{d}$  fits the data, that is,  $\hat{d}(\phi_n) \approx \tilde{p}_n$ ,  $n = 1, \dots, N$ ; and CO2)  $\hat{d}$  generalizes well to unseen data, i.e., if a new pair  $(\phi_{N+1}, \tilde{p}_{N+1})$  is received, then  $\hat{d}(\phi_{N+1}) \approx \tilde{p}_{N+1}$ . A popular approach to solve the aforementioned function learning problem is kernel-based learning, mainly due to its simplicity, universality, and good performance [33]. Furthermore, multiple works have demonstrated the merits of this framework for spectrum cartography; see Sec. B.1.

The first step when attempting to learn a function is to specify in which family of functions  $\hat{d}$  must be sought. In kernel-based learning, one seeks  $\hat{d}$  in a set known as a *reproducing-kernel Hilbert space* (RKHS), which is given by:

$$\mathcal{H} := \left\{ d : d(\boldsymbol{\phi}) = \sum_{i=1}^{\infty} \alpha_i \kappa(\boldsymbol{\phi}, \boldsymbol{\phi}'_i), \boldsymbol{\phi}'_i \in \mathcal{F}, \alpha_i \in \mathbb{R} \right\}, \quad (\text{B.5})$$

where  $\kappa : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$  is a symmetric and positive definite function known as *reproducing kernel* [34]. Although kernel methods can use any reproducing kernel, a common choice is the so-called Gaussian *radial basis function*  $\kappa(\boldsymbol{\phi}, \boldsymbol{\phi}') := \exp[-\|\boldsymbol{\phi} - \boldsymbol{\phi}'\|^2/(2\sigma^2)]$ , where  $\sigma > 0$  is a parameter selected by the user. As any Hilbert space,  $\mathcal{H}$  has an associated inner product and norm. For an RKHS function  $d(\boldsymbol{\phi}) = \sum_{i=1}^{\infty} \alpha_i \kappa(\boldsymbol{\phi}, \boldsymbol{\phi}'_i)$ , the latter is given by:

$$\|d\|_{\mathcal{H}}^2 := \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \alpha_i \alpha_j \kappa(\boldsymbol{\phi}'_i, \boldsymbol{\phi}'_j). \quad (\text{B.6})$$

Kernel-based learning typically solves a problem of the form:

$$\hat{d} = \arg \min_{d \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^N \mathcal{L}(\tilde{p}_n, \boldsymbol{\phi}_n, d(\boldsymbol{\phi}_n)) + \omega(\|d\|_{\mathcal{H}}), \quad (\text{B.7})$$

where  $\mathcal{L}$  is a loss function quantifying the deviation between the observations  $\{\tilde{p}_n\}_{n=1}^N$  and the predictions  $\{d(\boldsymbol{\phi}_n)\}_{n=1}^N$  returned by a candidate  $d$ ; and  $\omega$  is an increasing function. The first term in (B.7) promotes function estimates satisfying CO1. The second term promotes estimates satisfying CO2 by limiting overfitting. Intuitively,  $\|\cdot\|_{\mathcal{H}}$  captures a certain form of smoothness that limits the variability of  $d$ .

Although there exist different candidate functions for  $\mathcal{L}$  and  $\omega$  in kernel-base learning, typical choices are  $\mathcal{L}(\tilde{p}_n, \boldsymbol{\phi}_n, d(\boldsymbol{\phi}_n)) = (\tilde{p}_n - d(\boldsymbol{\phi}_n))^2$  and  $\omega(\|d\|_{\mathcal{H}}) = \lambda \|d\|_{\mathcal{H}}^2$ , where  $\lambda > 0$  is a regularization parameter that balances smoothness and goodness of fit. For this choice,  $\hat{d}$  is termed *kernel ridge regression* estimate [33, Ch. 4], and is the one used in our experiments for simplicity. The goal is therefore to solve (B.7). However, since  $\mathcal{H}$  is generally infinite dimensional, (B.7) cannot be directly solved. Fortunately, one can invoke the *representer theorem* [34], which states that the solution to (B.7) is of the form:

$$\hat{d}(\boldsymbol{\phi}) = \sum_{n=1}^N \alpha_n \kappa(\boldsymbol{\phi}, \boldsymbol{\phi}_n), \quad (\text{B.8})$$

for some  $\{\alpha_n\}_{n=1}^N$ . Although the representer theorem does not provide  $\{\alpha_n\}_{n=1}^N$ , these coefficients can be obtained by substituting (B.8) into (B.7) and solving the resulting problem with respect to them. Applying this procedure for kernel ridge regression results in the problem:

$$\hat{\boldsymbol{\alpha}} = \arg \min_{\boldsymbol{\alpha}} \frac{1}{N} \|\tilde{\boldsymbol{p}} - \mathbf{K}\boldsymbol{\alpha}\|^2 + \lambda \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}, \quad (\text{B.9})$$

where  $\boldsymbol{\alpha} := [\alpha_1, \dots, \alpha_N]^\top$ ,  $\tilde{\boldsymbol{p}} := [\tilde{p}_1, \dots, \tilde{p}_N]^\top$ , and  $\mathbf{K}$  is a positive-definite  $N \times N$  matrix whose  $(n, n')$ -th entry is  $\kappa(\boldsymbol{\phi}_n, \boldsymbol{\phi}_{n'})$ . Problem (B.9) can be readily solved in closed-form



as  $\hat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda N \mathbf{I}_N)^{-1} \tilde{\mathbf{p}}$ . The estimate  $\hat{d}$  solving (B.7) for kernel ridge regression can be recovered by substituting the resulting  $\{\alpha_n\}_{n=1}^N$  into (B.8). To obtain the predicted power of the C2M at a query location  $\mathbf{x}$  where the pilot signals are given by  $\mathbf{Y}$ , one just evaluates the LocF estimate  $\hat{p}_{\mathbf{Y}}(\mathbf{Y}) = \hat{d}(\boldsymbol{\phi}(\mathbf{Y}))$ .

## B.4 Location-Free Features

As described in Sec. B.3.1, LocB cartography algorithms learn a function of the location estimate. In the machine learning terminology, the *features* are the spatial coordinates of the sensor locations. On the other hand, the features used by LocF cartography are the entries of  $\boldsymbol{\phi}(\cdot)$ . In principle,  $\boldsymbol{\phi}(\mathbf{Y})$  could be set to contain the same features as the ones used by  $\hat{\mathbf{l}}(\cdot)$ ; see Sec. B.3. However, it is generally preferable to use features specifically tailored to LocF cartography. This section accomplishes the design of these features in several steps.

### B.4.1 Feature Extraction

In Sec. B.3.1,  $\boldsymbol{\phi}(\mathbf{Y})$  comprised  $M$  features used by typical localization algorithms, e.g. T(D)oA or DoA. The key observation is that, although these features are appropriate for localization, a different set of features may be preferable for LocF cartography. To come up with a natural feature design, this section first reviews the features used by typical localization algorithms (hence for LocB cartography) and analyzes their limitations. Inspired by this analysis, a novel feature extraction approach is proposed. To simplify the exposition, the scenario where sensors are synchronized with the base stations is presented first. A more practical setup, where this synchronization is not required, will be considered next.

#### B.4.1.1 Sensors are Synchronized with Base Stations

The received pilot signal is generally modeled as:

$$y_{l,n}[k] := a_l[k] * h_{l,n}[k] + w_{l,n}[k], \quad (\text{B.10})$$

where  $a_l[k]$  is the  $k$ -th sample of the  $l$ -th transmitted pilot signal,  $h_{l,n}[k]$  is the discrete-time channel impulse response between the  $l$ -th base station and the sensor at the  $n$ -th location, and  $w_{l,n}[k]$  is the noise term. The discrete-time impulse response  $h_{l,n}[k]$  is obtained next from its analog counterpart  $h_{l,n}(t)$ , which follows the conventional multipath channel model with  $P_{l,n}$  components:

$$h_{l,n}(t) = \sum_{p=1}^{P_{l,n}} \alpha_{l,n}^{(p)} \delta(t - t_{l,n}^{(p)}), \quad (\text{B.11})$$

where  $\delta(\cdot)$  is the Dirac delta distribution and  $\alpha_{l,n}^{(p)} \in \mathbb{R}$  and  $t_{l,n}^{(p)}$  are respectively the amplitude and delay of the  $p$ -th path. After up-conversion to the carrier frequency  $f_c$ , the pilot signal of the  $l$ -th base station is transmitted and received by the sensor at the

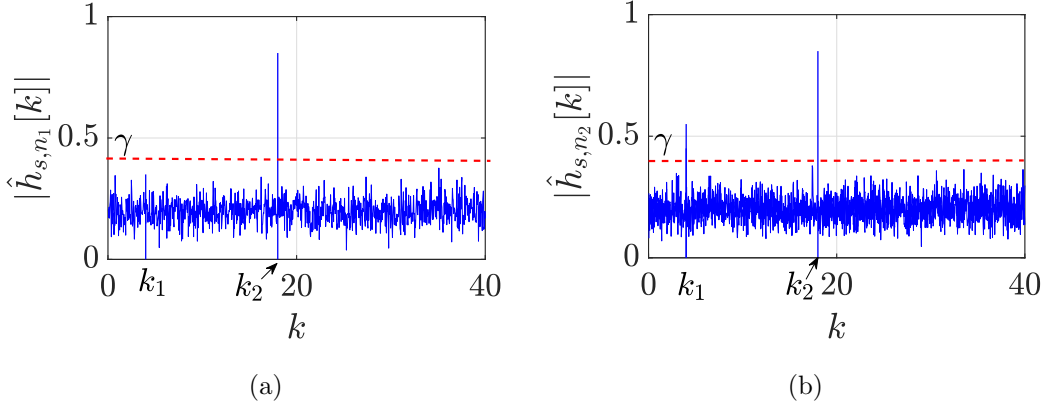


Figure B.3: Extraction of ToA from digital impulse response measured at two points that are spatially close. In (a), the ToA estimate is proportional to  $k_2$ ; whereas in (b), the ToA estimate is proportional to  $k_1$ .

$n$ -th measurement point, which bandpass-filters with bandwidth  $B$ , down-converts, and samples at the Nyquist rate  $T = 1/B$ . Therefore, the received noiseless samples are given by  $y_{l,n}[k]$  in (B.10), where [35, 36]:

$$h_{l,n}[k] = \sum_{p=1}^{P_{l,n}} \alpha_{l,n}^{(p)} e^{-j2\pi f_c t_{l,n}^{(p)}} \operatorname{sinc} \left( k - \frac{t_{l,n}^{(p)}}{T} \right). \quad (\text{B.12})$$

In view of these expressions, one of the most natural estimators for the ToA  $\tau_{l,n} := t_{l,n}^{(1)}$  is:

$$\hat{\tau}_{l,n} := T \cdot \min\{k : |\hat{h}_{l,n}[k]| \geq \gamma\}, \quad (\text{B.13})$$

where  $\hat{h}_{l,n}[k]$  is an estimate of  $h_{l,n}[k]$  and  $\gamma$  is typically set as a function of the signal-to-noise ratio [26].

It will be argued next that such a ToA feature does not evolve smoothly over space in presence of multipath, and therefore, this may negatively impact estimation performance, as occurs with LocB cartography; see discussion about Fig. B.1 in Sec. B.1. For simplicity, assume that  $a_l[k] = \delta[k]$ , where  $\delta[k]$  is the Kronecker delta. In this case, one can directly estimate  $h_{l,n}[k]$  as  $\hat{h}_{l,n}[k] = y_{l,n}[k] = h_{l,n}[k] + w_{l,n}[k]$ , which is a noisy version of  $h_{l,n}[k]$ . To see the impact of multipath, consider a simple example where the measurement points  $\mathbf{x}_{n_1}$  and  $\mathbf{x}_{n_2}$  lie close to each other and the channel impulse responses are given by  $\hat{h}_{l,n_1}[k] = \alpha_{l,n_1}^{(1)} \delta[k - k_{l,n_1}^{(1)}] + \alpha_{l,n_1}^{(2)} \delta[k - k_{l,n_1}^{(2)}] + w_{l,n_1}[k]$  and  $\hat{h}_{l,n_2}[k] = \alpha_{l,n_2}^{(1)} \delta[k - k_{l,n_2}^{(1)}] + \alpha_{l,n_2}^{(2)} \delta[k - k_{l,n_2}^{(2)}] + w_{l,n_2}[k]$ . Due to their spatial proximity, it follows that:

$$\alpha_{l,n_1}^{(1)} \approx \alpha_{l,n_2}^{(1)}, \quad \alpha_{l,n_1}^{(2)} \approx \alpha_{l,n_2}^{(2)}, \quad (\text{B.14a})$$

$$k_{l,n_1}^{(1)} \approx k_{l,n_2}^{(1)} \approx k_1, \quad k_{l,n_1}^{(2)} \approx k_{l,n_2}^{(2)} \approx k_2, \quad (\text{B.14b})$$

for some  $k_1$  and  $k_2$ . Assuming for simplicity that the effects of noise are negligible, if

$|\alpha_{l,n_1}^{(1)}| < \gamma < |\alpha_{l,n_1}^{(2)}|$  and  $\gamma < |\alpha_{l,n_2}^{(1)}|$ , then the ToA estimates are:

$$\begin{aligned}\hat{\tau}_{n_1} &:= T \cdot \min\{k : |\hat{h}_{l,n_1}[k]| \geq \gamma\} = Tk_{l,n_1}^{(2)} \approx Tk_2, \\ \hat{\tau}_{n_2} &:= T \cdot \min\{k : |\hat{h}_{l,n_2}[k]| \geq \gamma\} = Tk_{l,n_2}^{(1)} \approx Tk_1.\end{aligned}$$

This scenario is illustrated in Fig. B.3. Despite how close their locations and observed impulse responses are, the ToA estimates at locations  $\mathbf{x}_{n_1}$  and  $\mathbf{x}_{n_2}$  can be quite different, which establishes that the ToA estimate in (B.13) is not a smooth function of the spatial location.

Since this non-smoothness negatively affects the performance of the proposed LocF cartography estimator (and since the latter does not need ToA estimates that are proportional to the distance, as occurs in LocB cartography), a promising candidate for feature would be the *center of mass* (CoM) of the estimated impulse response:

$$\text{CoM}_{l,n} := \frac{\sum_{k=0}^{K-1} |\hat{h}_{l,n}[k]|^2 k}{\sum_{k=0}^{K-1} |\hat{h}_{l,n}[k]|^2},$$

where  $K$  is the number of samples. To see why such a feature evolves smoothly over space, suppose that the effects of noise are negligible and note that this CoM feature applied to the channel impulse responses in the previous example yields:

$$\begin{aligned}\text{CoM}_{l,n_1} &= \frac{k_{l,n_1}^{(1)} |\alpha_{l,n_1}^{(1)}|^2 + k_{l,n_1}^{(2)} |\alpha_{l,n_1}^{(2)}|^2}{|\alpha_{l,n_1}^{(1)}|^2 + |\alpha_{l,n_1}^{(2)}|^2}, \\ \text{CoM}_{l,n_2} &= \frac{k_{l,n_2}^{(1)} |\alpha_{l,n_2}^{(1)}|^2 + k_{l,n_2}^{(2)} |\alpha_{l,n_2}^{(2)}|^2}{|\alpha_{l,n_2}^{(1)}|^2 + |\alpha_{l,n_2}^{(2)}|^2}.\end{aligned}$$

From (B.14), it follows that  $\text{CoM}_{l,n_1} \approx \text{CoM}_{l,n_2}$ , which indicates that the CoM is indeed a feature that evolves smoothly over space, and therefore preferable for LocF cartography. In this case, the feature vector at the  $n$ -th sensor location becomes  $\phi_n = [\text{CoM}_{1,n}, \dots, \text{CoM}_{L,n}]^\top$ .

#### B.4.1.2 Sensors are not Synchronized with Base Stations

Since synchronization requires more expensive equipment and becomes challenging in multipath scenarios, TDoA estimates are generally preferred for localization. TDoA estimates are typically obtained by extracting the lag corresponding to the maximum cross-correlation of a pair of received pilot signals [37]. Assuming zero-mean, the cross-correlation between two pilot signals received by the sensor at the  $n$ -th location is defined as:

$$c_{l,l',n}[i] := \mathbb{E}\{y_{l,n}[k]y_{l',n}^*[k-i]\} \quad \text{with } l \neq l'. \quad (\text{B.15})$$

With  $a_l[k] = a_{l'}[k]$  a white process with power  $\sigma_a^2$  and uncorrelated with  $w_{l,n}[k]$  and  $w_{l',n}[k]$ , also uncorrelated with each other, it can be easily seen that:

$$c_{l,l',n}[i] = \sigma_a^2 (h_{l,n}[i] * h_{l',n}^*[-i]).$$

A common estimate of the TDoA  $\Delta_{l,l',n}$  is (see e.g. [37]):

$$\hat{\Delta}_{l,l',n} = T \cdot \arg \max_i \{ |\hat{c}_{l,l',n}[i]| \}, \quad (\text{B.16})$$

where  $\hat{c}_{l,l',n}[i]$  is an estimate of  $c_{l,l',n}[i]$ . To see the intuition behind this estimator, note that  $\hat{h}_{l,n}[k] = \alpha_{l,n}^{(1)} \delta[k - k_{l,n}^{(1)}]$  and  $\hat{h}_{l',n}[k] = \alpha_{l',n}^{(1)} \delta[k - k_{l',n}^{(1)}]$  in a free-space channel with large bandwidth  $B$ . This implies that:

$$\begin{aligned} c_{l,l',n}[i] &= \sigma_a^2 \alpha_{l,n}^{(1)} \left( \alpha_{l',n}^{(1)} \right)^* \delta \left[ i - \left( k_{l,n}^{(1)} - k_{l',n}^{(1)} \right) \right] \\ &= \sigma_a^2 \alpha_{l,n}^{(1)} \left( \alpha_{l',n}^{(1)} \right)^* \delta \left[ i - \Delta_{l,l',n}/T \right], \end{aligned}$$

and therefore the lag of the maximum magnitude of  $c_{l,l',n}[i]$  provides the TDoA in this simple scenario.

Similar arguments to those used in Sec. B.4.1.1 to conclude that the ToA estimates are not spatially smooth can also be invoked to reach the same conclusion for TDoA. Likewise, following the same rationale as in Sec. B.4.1.1, this section proposes alleviating the aforementioned issue by adopting features of the form:

$$\text{CoM}_{l,l',n} := \frac{\sum_{i=-K+1}^{K-1} |c_{l,l',n}[i]|^2 i}{\sum_{i=-K+1}^{K-1} |c_{l,l',n}[i]|^2}, \quad (\text{B.17})$$

where  $\text{CoM}_{l,l',n}$  is the CoM of the cross-correlation between the  $l$ -th and  $l'$ -th pilot signals. The proposed feature has three advantages: i) it is smooth, as portrayed later in Sec. B.5.1, ii) it does not require synchronization between the localization base stations and the sensors, and iii) it does not require the knowledge of the impulse responses. With this choice, the feature vector at the  $n$ -th measurement location becomes:

$$\begin{aligned} \phi_n &= [\text{CoM}_{1,2,n}, \text{CoM}_{1,3,n}, \dots, \text{CoM}_{1,L,n}, \\ &\quad \text{CoM}_{2,3,n}, \dots, \text{CoM}_{L-1,L,n}]^\top. \end{aligned} \quad (\text{B.18})$$

## B.4.2 Cartography from a Reduced Set of Features

As argued earlier in Sec. B.3.1, learning becomes difficult when the number of input features  $M$  is high. This section develops a scheme to reduce this number of features to improve estimation performance in LocF cartography.

As stated in the previous section, in LocB cartography, the feature vectors correspond to the coordinates of the estimated location. Application of the localization algorithm represented by the function  $\hat{\mathbf{l}}$  in (B.3) naturally reduces dimensionality from the original  $M$  features to just 2 or 3. On the other hand, in the case of LocF cartography, a larger number  $N$  of measurements to learn  $\hat{d}$  in (B.4) may be necessary to attain a target accuracy if  $M$  is large. This observation calls for a dimensionality reduction step that condenses the information of the feature vectors  $\{\phi_n\}_{n=1}^N \subset \mathbb{R}^M$  into vectors  $\{\bar{\phi}_n\}_{n=1}^N \subset \mathbb{R}^r$  of a reduced size  $r$ . Intuitively,  $r$  should be the minimum number that preserves most information while eliminating most of the noise in  $\{\phi_n\}_{n=1}^N$ . Even if some information is lost, the reduction in the error entailed by the fact that the function to be estimated has fewer input arguments may pay off in practice.

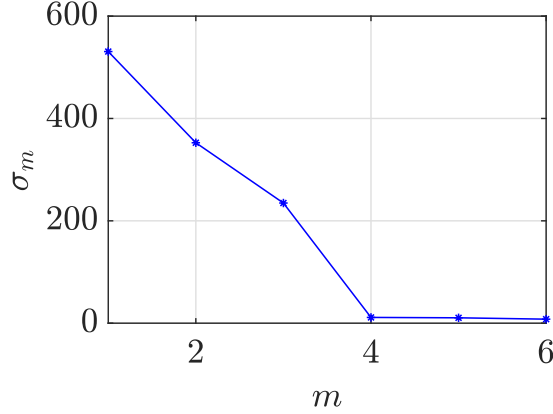


Figure B.4: Singular values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$  of  $\Phi$  in non-increasing order for a multipath environment with  $L = 4$  transmitters.

In the cases where the feature vectors  $\{\phi_n\}_{n=1}^N$  lie close to a low-dimensional subspace, the coordinates of these vectors with respect to a basis for such a subspace may constitute a suitable reduced set of features. To see this, it is instructive to start by considering the scenario of TDoA features. Suppose, for simplicity, that the effects of noise are negligible, so that the TDoA estimates  $\hat{\Delta}_{l,l',n}$  approximately equal the true TDoAs  $\Delta_{l,l',n}$ . Then, the rows of  $\Phi := [\phi_1, \dots, \phi_N]$  are of the form  $\Delta_{l,l'} := [\Delta_{l,l',1}, \Delta_{l,l',2}, \dots, \Delta_{l,l',N}]^\top$ . If  $\tau_l := [\tau_{l,1}, \dots, \tau_{l,N}]^\top$  collects the ToA from the  $l$ -th base station to all sensor locations, then it clearly holds that  $\Delta_{l,l'} = \tau_l - \tau_{l'}$ . Consequently,  $\Delta_{1,l} - \Delta_{1,l'} = \tau_1 - \tau_l - (\tau_1 - \tau_{l'}) = \tau_{l'} - \tau_l = \Delta_{l',l}$ , which implies that all rows of  $\Phi$  are linear combinations of the  $L - 1$  rows  $\{\Delta_{1,l}\}_{l=2}^L$ . Thus, the rank of  $\Phi$  is at most  $L - 1$  or, equivalently, the vectors  $\{\phi_n\}_{n=1}^N$  lie in a subspace of dimension  $L - 1$ . When effects of noise are noticeable, one would expect that the vectors  $\{\phi_n\}_{n=1}^N$  lie *close* to a subspace of dimension  $L - 1$ .

Similarly, one can expect that when the entries of the vectors  $\{\phi_n\}_{n=1}^N$  are given by (B.17), these vectors also lie close to a low-dimensional subspace since CoM features are proportional to the TDoAs in absence of multipath; see Sec. B.4.1. This phenomenon can be illustrated through simulation (see Sec. B.5 for more details). Fig. B.4 depicts the singular values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$  of  $\Phi$  in non-increasing order for a multipath environment described in Sec. B.5 with  $L = 4$ . As expected, roughly  $r = L - 1 = 3$  directions capture almost all the energy of the rows of  $\Phi$ .

When a set of random vectors lie close to a subspace, an appealing approach for dimensionality reduction is principal component analysis (PCA) [31, Ch. 12], which obtains the reduced feature vectors by projecting the input data vectors onto the subspace that preserves most of the energy. Since in this paper no probabilistic assumptions have been introduced on  $\{\phi_n\}_{n=1}^N$ , the typical formulation of PCA is not directly applicable. However, as detailed next, it is not difficult to extend this idea to the fully deterministic scenario, which furthermore provides intuition.

Assume w.l.o.g. a centered set of feature vectors, i.e.,  $(1/N) \sum_{n=1}^N \phi_n = \mathbf{0}$ . If not centered, just subtract the mean by replacing  $\Phi$  with  $\Phi - (1/N)\Phi\mathbf{1}\mathbf{1}^\top$ . The subspace that captures most of the energy of the observations can be determined using the singular

value decomposition (SVD) of  $\Phi$ , which for  $M < N$  is given by:

$$\Phi = [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_2 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^\top \\ \mathbf{V}_2^\top \end{bmatrix}, \quad (\text{B.19})$$

where  $\Sigma_1 := \text{diag}\{\sigma_1, \dots, \sigma_r\}$  contains the  $r$  largest singular values of  $\Phi$ ,  $\Sigma_2 := \text{diag}\{\sigma_{r+1}, \dots, \sigma_M\}$  contains the  $M - r$  smallest, and the columns of  $\mathbf{U} := [\mathbf{U}_1 \quad \mathbf{U}_2]$  (respectively  $\mathbf{V} := [\mathbf{V}_1, \mathbf{V}_2]$ ) are the left (right) singular vectors of  $\Phi$ . Clearly, if the data vectors  $\{\phi_n\}_{n=1}^N$  are multiplied by the orthogonal matrix  $\mathbf{U}^\top$ , the resulting vectors  $\{\phi'_n\}_{n=1}^N$ , with  $\phi'_n := \mathbf{U}^\top \phi_n$ , contain the same information. Thus, one can replace  $\Phi$  with  $\Phi' := \mathbf{U}^\top \Phi$ .

By applying this transformation, which can be thought of as a generalized rotation, most of the energy of  $\Phi'$  is concentrated in its first  $r$  rows. To see this, note that the energy of the first  $r$  rows of  $\Phi'$  is given by:

$$\begin{aligned} \|\mathbf{U}_1^\top \Phi\|_F^2 &= \|\Sigma_1 \mathbf{V}_1^\top\|_F^2 = \text{Tr}(\Sigma_1 \mathbf{V}_1^\top \mathbf{V}_1 \Sigma_1^\top) \\ &= \text{Tr}(\Sigma_1 \Sigma_1^\top) = \|\Sigma_1\|_F^2 = \sum_{m=1}^r \sigma_m^2, \end{aligned}$$

whereas the energy of the last  $M - r$  rows of  $\Phi'$  is given by:

$$\|\mathbf{U}_2^\top \Phi\|_F^2 = \|\Sigma_2\|_F^2 = \sum_{m=r+1}^M \sigma_m^2.$$

When  $r = L - 1$ , since the rows of  $\Phi$  lie approximately in a subspace of dimension  $r$ , it follows that  $\sigma_m \approx 0$  for  $m > r$ . Therefore  $\sum_{m=1}^r \sigma_m^2 \gg \sum_{m=r+1}^M \sigma_m^2$  and, hence,  $\|\mathbf{U}_1^\top \Phi\|_F^2 \gg \|\mathbf{U}_2^\top \Phi\|_F^2$ . Equivalently, most of the energy of the vectors  $\{\phi'_n\}_{n=1}^N$  is concentrated in their first  $r$  entries. This observation suggests using the first  $r$  entries of the vectors  $\{\phi'_n\}_{n=1}^N$  as features, while discarding the rest. That is, the reduced dimensionality feature vectors will be given by  $\{\bar{\phi}_n\}_{n=1}^N$ , where  $\bar{\phi}_n := \mathbf{U}_1^\top \phi_n$ . Note that  $\bar{\phi}_n$  is just the vector of coordinates of  $\phi_n$  with respect to the basis composed of the columns of  $\mathbf{U}_1$ .

The number  $r$  of entries of the new feature vectors  $\{\bar{\phi}_n\}_{n=1}^N$  may be potentially much smaller than  $M$  and can therefore boost estimation performance meaningfully. For instance, when  $\{\phi_n\}_{n=1}^N$  are given by (B.18), this reduction is from  $M = L(L - 1)/2$  features to  $r = L - 1$  features.

In scenarios of very strong multipath, the rows of  $\Phi$  may not lie close to any subspace of dimension  $L - 1$ . In those cases, it may be worth choosing a value of  $r$  greater than  $L - 1$ . A possibility is to specify a fraction  $\eta \in [0, 1]$  of the energy of  $\Phi$  that must be kept in  $\bar{\Phi} := \mathbf{U}_1^\top \Phi$ , and choose  $r$  to be the smallest integer that guarantees this condition, that is:

$$r = \min \left\{ r' : \frac{\sum_{\bar{m}=1}^{r'} \sigma_{\bar{m}}^2}{\sum_{m=1}^M \sigma_m^2} \geq \eta \right\}. \quad (\text{B.20})$$

To summarize, the problem of LocF cartography with the technique for reducing the set of features introduced in this section is as follows. Given the original set of measurements  $\{\phi_n\}_{n=1}^N \subset \mathbb{R}^M$ , one must form the matrix  $\Phi$ , compute  $\mathbf{U}_1$  from the SVD in (B.19), and

obtain the reduced features  $\{\bar{\phi}_n\}_{n=1}^N \subset \mathbb{R}^r$  where  $\bar{\phi}_n = \mathbf{U}_1^\top \phi_n$ . Then, the function  $\hat{d}$  is obtained from the pairs  $\{(\bar{\phi}_n, \tilde{p}_n)\}_{n=1}^N$  using the approach in Sec. B.3.2. To evaluate the resulting map at a query location where the received pilot signals are given by  $\mathbf{Y}$ , one must simply obtain  $\hat{d}(\mathbf{U}_1^\top \phi(\mathbf{Y}))$ .

### B.4.3 Dealing with Missing Features

Due to propagation effects, the signal-to-noise ratio of some of the received pilot signals may be too low for feature extraction. In this case, the features associated with those pilot signals may be unreliable or simply unavailable. This section develops techniques to cope with such missing features.

Let  $\Omega \subset \{1, \dots, M\} \times \{1, \dots, N\}$  be such that  $(m, n) \in \Omega$  iff the  $m$ -th feature is available at the  $n$ -th measurement location and define the ‘‘incomplete’’ feature matrix  $\check{\Phi} \in (\mathbb{R} \cup \{\text{FiM}\})^{M \times N}$  as:

$$(\check{\Phi})_{m,n} = \begin{cases} (\phi_n)_m + \varsigma_{m,n} & \text{if } (m, n) \in \Omega \\ \text{FiM} & \text{otherwise,} \end{cases} \quad (\text{B.21})$$

where  $\varsigma_{m,n}$  explicitly models error in the feature extraction and the symbol FiM represents that the corresponding feature is missing. Since the matrix  $\check{\Phi}$  contains missing features, the LocF cartography scheme presented so far is not directly applicable. The missing features must be filled first. Hence, the goal is, given  $\check{\Phi}$ , find  $\Phi \in \mathbb{R}^{M \times N}$  that agrees with  $\check{\Phi}$  on  $\Omega$ . A popular approach to address such a matrix completion task is via rank minimization [38]:

$$\begin{aligned} & \underset{\Phi}{\text{minimize}} \quad \text{rank}(\Phi) \\ & \text{subject to} \quad \mathcal{P}_\Omega(\Phi) = \mathcal{P}_\Omega(\check{\Phi}), \end{aligned} \quad (\text{B.22})$$

where

$$\begin{aligned} \mathcal{P}_\Omega : (\mathbb{R} \cup \{\text{FiM}\})^{M \times N} &\longrightarrow \mathbb{R}^{M \times N} \\ \check{\Phi} &\longmapsto \mathcal{P}_\Omega(\check{\Phi}), \end{aligned}$$

with

$$\left(\mathcal{P}_\Omega(\check{\Phi})\right)_{m,n} = \begin{cases} (\check{\Phi})_{m,n} & \text{if } (m, n) \in \Omega \\ 0 & \text{if } (m, n) \notin \Omega. \end{cases}$$

Although this problem is non-convex, efficient solvers exist based on convex relaxation [39, 40]. A legitimate question would be what is the minimum number of available features required to recover a reasonable reconstruction of  $\Phi$ . As a guideline, a result in [41] establishes that, under certain conditions, the minimum number of available features to recover  $\Phi \in \mathbb{R}^{M \times N}$  is  $\mathcal{O}\left(\tilde{N} \text{rank}(\Phi) \log(\tilde{N})\right)$  where  $\tilde{N} = \max(M, N)$ .

Although the aforementioned rank minimization approach could, in principle, be used, it suffers from two limitations. First, it does not exploit the prior information that  $\Phi$  can be well approximated by a matrix of rank  $r$ , where  $r$  is typically  $L - 1$ ; see Fig. B.4.

Second, the constraint in (B.22) would render the reconstructed matrix sensitive to the noise  $\{\varsigma_{m,n}\}_{m,n}$  present in  $\check{\Phi}$ . Thus, an appealing alternative to (B.22) would be:

$$\begin{aligned} \mathring{\Phi} := \underset{\Phi}{\operatorname{argmin}} \quad & \frac{1}{2} \|\mathcal{P}_\Omega(\Phi) - \mathcal{P}_\Omega(\check{\Phi})\|_F^2 \\ \text{subject to} \quad & \Phi \in \mathcal{M}_r, \end{aligned} \quad (\text{B.23})$$

where  $\mathcal{M}_r := \{\Phi \in \mathbb{R}^{M \times N} : \operatorname{rank}(\Phi) = r\}$  is the smooth manifold of  $r$ -rank  $M \times N$  matrices.

There exist algorithms to find local minima of the non-convex problem (B.23). One example based on manifold optimization [42] is the linear retraction-based geometric conjugate gradient (LRGeomCG) method from [43]. A less computationally expensive alternative is the singular value projection (SVP) method in [44], which is based on the traditional projected subgradient descent method.

After solving (B.23), all the columns of  $\mathring{\Phi} := [\mathring{\phi}_1, \dots, \mathring{\phi}_N]$  clearly lie in a subspace of dimension  $r$ . From the arguments in Sec. B.4.2, learning the map can be improved by suppressing this redundancy. To this end, one could use the technique in Sec. B.4.2, which would obtain the reduced-dimensionality feature vectors as follows:

$$\bar{\Phi} := [\bar{\phi}_1, \dots, \bar{\phi}_N] = \mathring{U}_1^\top \mathring{\Phi}. \quad (\text{B.24})$$

Here, the columns of  $\mathring{U}_1$  are the left singular vectors corresponding to the  $r$  largest singular values of  $\mathring{\Phi}$ . Nevertheless, since  $\mathring{\Phi}$  has rank  $r$ , it is not necessary to obtain  $\mathring{U}_1$  by means of an SVD. Namely, the columns of  $\mathring{U}_1$  can be directly obtained by orthonormalizing the first  $r$  linearly independent columns of  $\mathring{\Phi}$ , e.g. through Gram-Schmidt.

To sum up, to estimate a map using the proposed LocF cartography in presence of missing features is as follows. First, matrix  $\check{\Phi}$  is formed with the available features. Then, the completed matrix  $\mathring{\Phi}$  is obtained using LRGeomCG or SVP. Next,  $\mathring{U}_1$  is obtained through Gram-Schmidt over this completed matrix. Finally, one learns  $\hat{d}$  from  $\{(\bar{\phi}_n, \tilde{p}_n)\}_{n=1}^N$ , where  $\bar{\phi}_n$  is the  $n$ -th column of  $\bar{\Phi}$  in (B.24), using the approach in Sec. B.3.2.

To evaluate the estimated map at a test location, one would require in principle the feature vector  $\phi \in \mathbb{R}^M$  at that location or, alternatively, its reduced-dimensionality version  $\bar{\phi} \in \mathbb{R}^r$ . However, due to the phenomena described earlier, only some of the features of  $\phi$  may be available, which can be collected in the vector  $\check{\phi} \in (\mathbb{R} \cup \{\text{FiM}\})^M$ . The problem now is to find the reduced-dimensionality feature vector  $\bar{\phi}$  given  $\check{\phi}$ .

Since the columns of  $\mathring{\Phi}$  lie in an  $r$ -dimensional subspace for which the columns of  $\mathring{U}_1$  form an orthonormal basis, it is reasonable to say that the feature vector at the testing point  $\phi \in \mathbb{R}^M$  also lies in that subspace, meaning that this vector can be written as  $\phi = \mathring{U}_1 \bar{\phi}$  for some  $\bar{\phi}$ . The procedure to recover  $\bar{\phi}$  depends on whether  $\check{\phi}$  contains enough observed features. Let  $\Omega' \subset \{1, \dots, M\}$  be such that the  $m \in \Omega'$  iff the  $m$ -th feature is available in  $\check{\phi}$ . If  $M := |\Omega'| \geq r$ , one can think of finding  $\bar{\phi}$  using the well-known regularized least squares (RLS) method as:

$$\begin{aligned} \hat{\phi} = \arg \min_{\bar{\phi}} \quad & \left\| \mathcal{P}_{\Omega'}(\check{\phi}) - \mathcal{P}_{\Omega'}(\mathring{U}_1 \bar{\phi}) \right\|^2 \\ & + \mu (\bar{\phi} - \bar{\phi}_{\text{avg}})^\top \mathbf{C}^{-1} (\bar{\phi} - \bar{\phi}_{\text{avg}}), \end{aligned} \quad (\text{B.25})$$



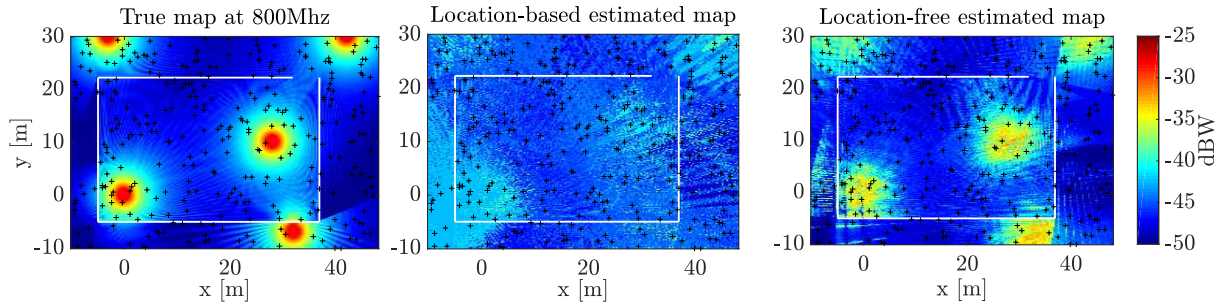


Figure B.5: (left) True map, (middle) LocB ( $\lambda' = 3.3 \times 10^{-3}$ ,  $\sigma' = 0.5$  m), and (right) LocF ( $\lambda = 1.9 \times 10^{-4}$ ,  $\sigma = 37$  m) estimated maps;  $N = 300$ ,  $L = 5$ ,  $B = 20$  MHz, and  $K = 10$ . The black crosses indicate the sensor locations and the solid white lines represent the walls of the building.

where

$$\mathcal{P}_{\Omega'} : (\mathbb{R} \cup \{\text{FiM}\})^M \longrightarrow \mathbb{R}^M$$

$$\check{\phi} \longmapsto \phi, (\phi)_m = \begin{cases} (\check{\phi})_m & \text{if } m \in \Omega' \\ 0 & \text{if } m \notin \Omega', \end{cases}$$

$\mu > 0$  is a regularization parameter,  $\bar{\phi}_{\text{avg}}$  and  $\mathbf{C} \in \mathbb{R}^{r \times r}$  are respectively the sample mean vector and covariance matrix of the coordinates of the completed features in the training phase, that is,  $\bar{\phi}_{\text{avg}} = (1/N)\bar{\Phi}\mathbf{1}$  and  $\mathbf{C} = (1/N)(\bar{\Phi} - \bar{\phi}_{\text{avg}}\mathbf{1}^\top)(\bar{\Phi} - \bar{\phi}_{\text{avg}}\mathbf{1}^\top)^\top$ . To solve Problem (B.25), let the elements of  $\Omega'$  be denoted as  $\Omega' := \{m_1, \dots, m_{\check{M}}\}$ . Then:

$$\hat{\phi} = \left( \mathring{\mathbf{U}}_1^\top \mathbf{S}^\top \mathbf{S} \mathring{\mathbf{U}}_1 + \mu \mathbf{C}^{-1} \right)^{-1} \left( \mathring{\mathbf{U}}_1^\top \mathbf{S}^\top \mathbf{S} \mathcal{P}_{\Omega'}(\check{\phi}) + \mu \mathbf{C}^{-1} \bar{\phi}_{\text{avg}} \right), \quad (\text{B.26})$$

where  $\mathbf{S} \in \{0, 1\}^{\check{M} \times M}$  is a row selection matrix with all entries equal to zero except for the entries  $(1, m_1), \dots, (\check{M}, m_{\check{M}})$ , which equal to 1. Thus,  $\mathbf{S} \mathcal{P}_{\Omega'}(\check{\phi}) = \mathbf{S} \mathring{\mathbf{U}}_1 \bar{\phi}$ . On the other hand, if  $\check{M} := |\Omega'| < r$ , it is not possible to identify  $\bar{\phi}$  from  $\check{\phi}$ . The extreme case would be when  $\check{M} = 0$ . A natural estimate at such point can be the spatial average of the signal power  $(1/N) \sum_n \tilde{p}_n$ .

## B.5 Numerical tests

This section evaluates the performance of LocF cartography in presence of multipath, where localization algorithms cannot achieve accurate location estimates. To this end, the simulations are carried out in a  $42 \times 27$  m structure comprising several parallel vertical planes modeling the external and internal walls of a building, the latter is located in a  $60 \times 40$  m rectangular area  $\mathcal{X}$ .

All the experiments described in this paper can be reproduced with the MATLAB code which is available at the first author's homepage.

The simulation area contains  $L$  active transmitters. Some of these are positioned inside the building, others outside. Matrix  $\mathbf{Y}_n \in \mathbb{C}^{L \times K}$  containing the noisy received

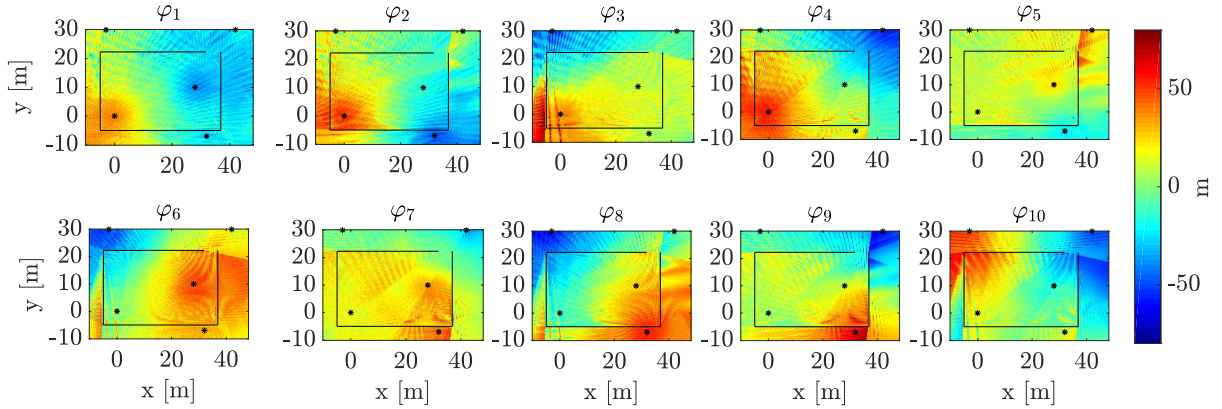


Figure B.6: Maps of the  $M = 10$  LocF features with  $L = 5$ ,  $B = 20$  MHz, and  $K = 10$ . The solid black lines represent the walls of the building and the black stars represent the transmitter locations.

Table B.1: Parameters used for the experiment in Fig. B.8.

|      |               |                       |                      |                      |                      |
|------|---------------|-----------------------|----------------------|----------------------|----------------------|
|      | $B$ (MHz)     | 50                    | 100                  | 200                  | 700                  |
|      | $K$           | 25                    | 50                   | 100                  | 350                  |
| LocB | $\sigma'$ (m) | 10.1                  | 8.9                  | 9                    | 7                    |
|      | $\lambda'$    | $1.8 \times 10^{-3}$  | $9.1 \times 10^{-4}$ | $7.1 \times 10^{-4}$ | $2.1 \times 10^{-4}$ |
| LocF | $\sigma$ (m)  | 27                    | 41                   | 53                   | 28                   |
|      | $\lambda$     | $3.81 \times 10^{-4}$ | $6.1 \times 10^{-5}$ | $1.1 \times 10^{-5}$ | $5 \times 10^{-4}$   |

pilot signals is generated according to (B.10), where  $K$  is adjusted depending on  $B$  to capture all the multipath components. For simplicity, the pilot signals are given by<sup>2</sup>  $a_l[k] = \delta[k]$  which implies that the rows of  $\mathbf{Y}_n \in \mathbb{C}^{L \times K}$  contain the impulse responses of the bandlimited channels between the  $L$  transmitters and the  $n$ -th measurement location. The channel  $h_{l,n}[k]$  is generated following (B.12) with a carrier frequency of 800 MHz and pilot channel bandwidth  $B = 1/T$ . The noise samples  $w_{l,n}[k]$  are independent normal random variables with zero-mean and variance -70 dBm. Propagation adheres to the *Motley-Keenan multi-wall radio propagation model* [45], which accounts for the direct path, up to 5 first-order wall reflections, and up to 5 wall-to-wall second-order reflections. Remarkably, the model captures the impact of the angle of incidence on the power of the reflected ray. For simplicity, the C2M is chosen to be the channel where localization pilot signals are transmitted. In practice, this is the case in the downlink of a cellular communication system such as LTE where the base stations transmit both communication signals and localization pilots.

To ensure that the measurements are obtained in the far-field propagation region, sensor locations are spread uniformly at random over  $\bar{\mathcal{X}}$ , which comprises those points in  $\mathcal{X}$  lying at least 3 wavelengths away from all transmitters. Note that, although the number of sensor locations is sometimes in the order of hundreds, this does not mean that a large

<sup>2</sup>Amplitude units are such that a signal  $x[k] = 1, \forall k$ , has power 1 W.

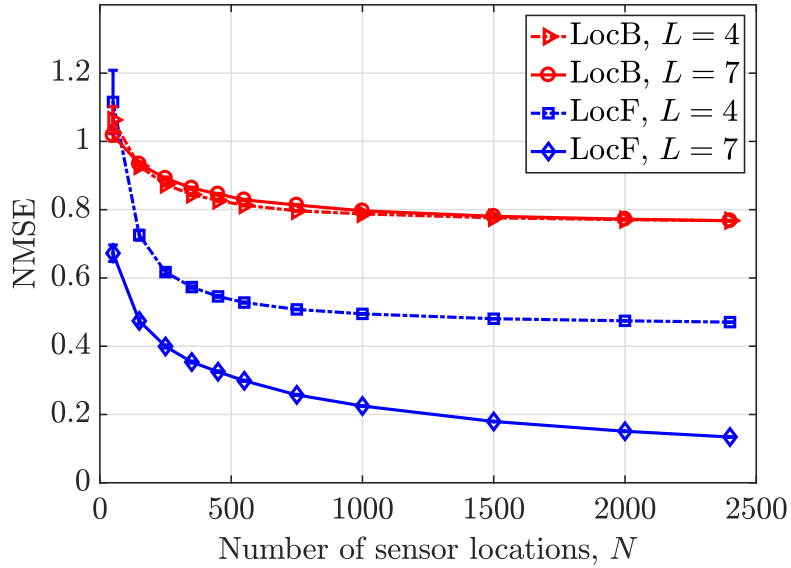


Figure B.7: Performance comparison between the LocF cartography ( $\lambda = 1.9 \times 10^{-4}$ ,  $\sigma = 37$  m) and the LocB cartography ( $\lambda' = 3.3 \times 10^{-3}$ ,  $\sigma' = 0.5$  m) with  $B = 20$  MHz and  $K = 10$ .

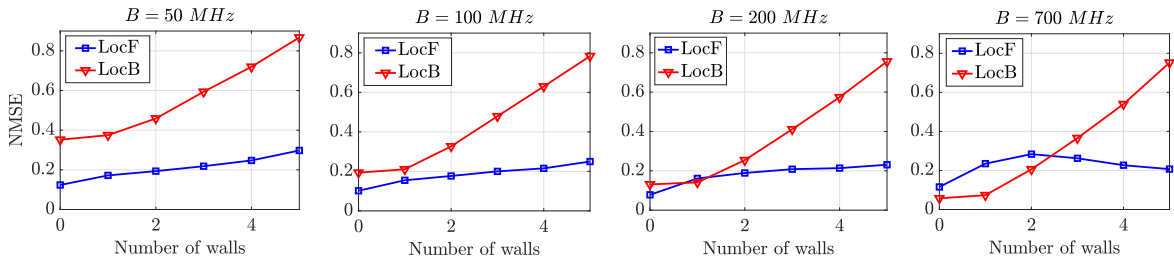


Figure B.8: LocF and LocB map NMSE as a function of the number of walls for different values of the bandwidth,  $L = 5$ ,  $N = 300$ .

number of sensing devices must be used since each device may gather measurements at tens or hundreds of spatial locations. The power measurement  $p_n$  (measured in dBW) of the C2M at position  $\mathbf{x}_n$  is corrupted by additive noise  $\epsilon_n$  to yield  $\tilde{p}_n = p_n + \epsilon_n$ , where  $\{\epsilon_n\}_{n=1}^N$  are independent normal random variables with zero-mean and variance  $\sigma_\epsilon^2$ . This variance is such that the signal-to-noise ratio defined as  $10 \log_{10}(\bar{p}^2/\sigma_\epsilon^2) \approx 40$  dB, where  $\bar{p} := \int_{\bar{\mathcal{X}}} p(\mathbf{x}) d\mathbf{x} / \int_{\bar{\mathcal{X}}} d\mathbf{x}$  is the spatial average of  $p(\mathbf{x})$ . This SNR is considered practical since the measurement noise power  $\sigma_\epsilon^2$  can be driven arbitrarily close to zero in practice by averaging over a sufficiently long time window.

Quantitative evaluation will compare the normalized mean square error (NMSE) defined as:

$$\text{NMSE} = \frac{\mathbb{E}\{|p(\mathbf{x}) - \hat{p}_{\mathbf{Y}}(\mathbf{Y}(\mathbf{x}), \mathcal{T})|^2\}}{\mathbb{E}\{|p(\mathbf{x}) - \bar{p}|^2\}}, \quad (\text{B.27})$$

where  $\hat{p}_{\mathbf{Y}}(\mathbf{Y}(\mathbf{x}), \mathcal{T})$  (measured in dBW) denotes the result of evaluating the map constructed from the training set  $\mathcal{T} := \{(\mathbf{Y}_n, \tilde{p}_n)\}_{n=1}^N$  at the location  $\mathbf{x}$ , where  $\mathbf{Y}(\mathbf{x})$  comprises

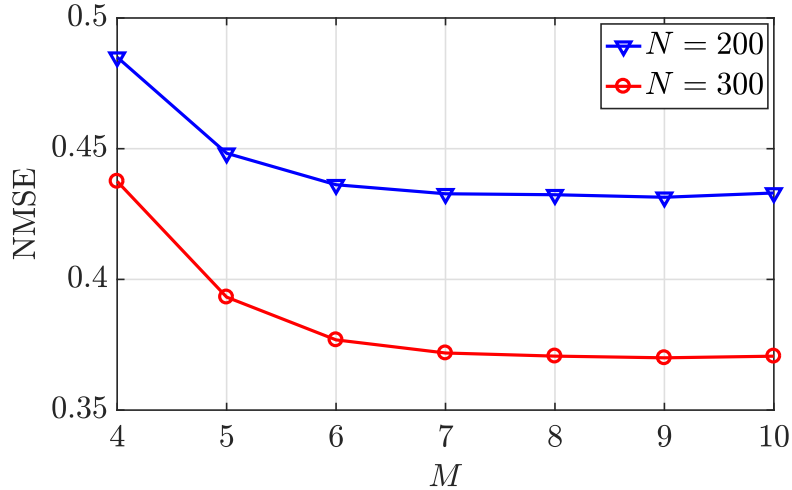


Figure B.9: LocF estimated map NMSE for different values of number of features  $M$  and sensor locations  $N$ , with  $L = 5$ ,  $B = 20$  MHz,  $K = 10$ ,  $\lambda = 1.9 \times 10^{-4}$ , and  $\sigma = 37$  m.

the received pilot signals at  $\mathbf{x}$ . The denominator in (B.27) normalizes the square error of the considered algorithm by the error incurred by the best data-agnostic estimator, which estimates the spatial average  $\bar{p}$  at all points. Thus, the adopted performance metric is higher than traditional NMSE, meaning that it is more challenging to obtain lower values. Furthermore  $\mathbb{E}\{\cdot\}$  denotes the expectation over the sensor locations and noise.

### B.5.1 LocF vs. LocB

To avoid the need for synchronization between transmitters and sensors, the LocF algorithm utilizes the features in (B.17), which additionally provide robustness to multipath and evolve smoothly over space; see Sec. B.4.1. Since this center of mass can be thought of as a lag, it is scaled by the sampling period  $T$  and speed of light  $c$  to obtain the corresponding range difference, i.e.:

$$\phi_n := Tc [\text{CoM}_{1,2,n}, \text{CoM}_{1,3,n}, \dots, \text{CoM}_{1,L,n}, \text{CoM}_{2,3,n}, \dots, \text{CoM}_{L-1,L,n}]^\top. \quad (\text{B.28})$$

Using these features, the LocF algorithm uses the kernel ridge regression technique in Sec. B.3.2 with Gaussian radial basis functions with parameter  $\sigma$ . The reason is that this *universal kernel* is capable of approximating arbitrary continuous functions that vanish at infinity [46]. On the other hand, for LocB cartography, the feature vector  $\phi_n = \hat{\mathbf{x}}_n \in \mathbb{R}^2$  comprises estimates of the spatial coordinates of the  $n$ -th sensor location obtained by the *iterative re-weighting squared range difference-least squares* (IRWSRD-LS) algorithm [47], which features state-of-the-art localization performance. This algorithm is applied over TDoA features extracted from  $\{\mathbf{Y}_n\}_{n=1}^N$  through (B.16). At the  $n$ -th sensor location, these features  $\{\hat{\Delta}_{1,l',n}\}_{l'=2}^L$  comprise the TDoA between a reference base station and the remaining  $L-1$  base stations. Enlarging this set by including TDoA measurements  $\hat{\Delta}_{l,l',n}$  with  $l \neq 1$  would not be beneficial for the estimation performance as discussed in [48].

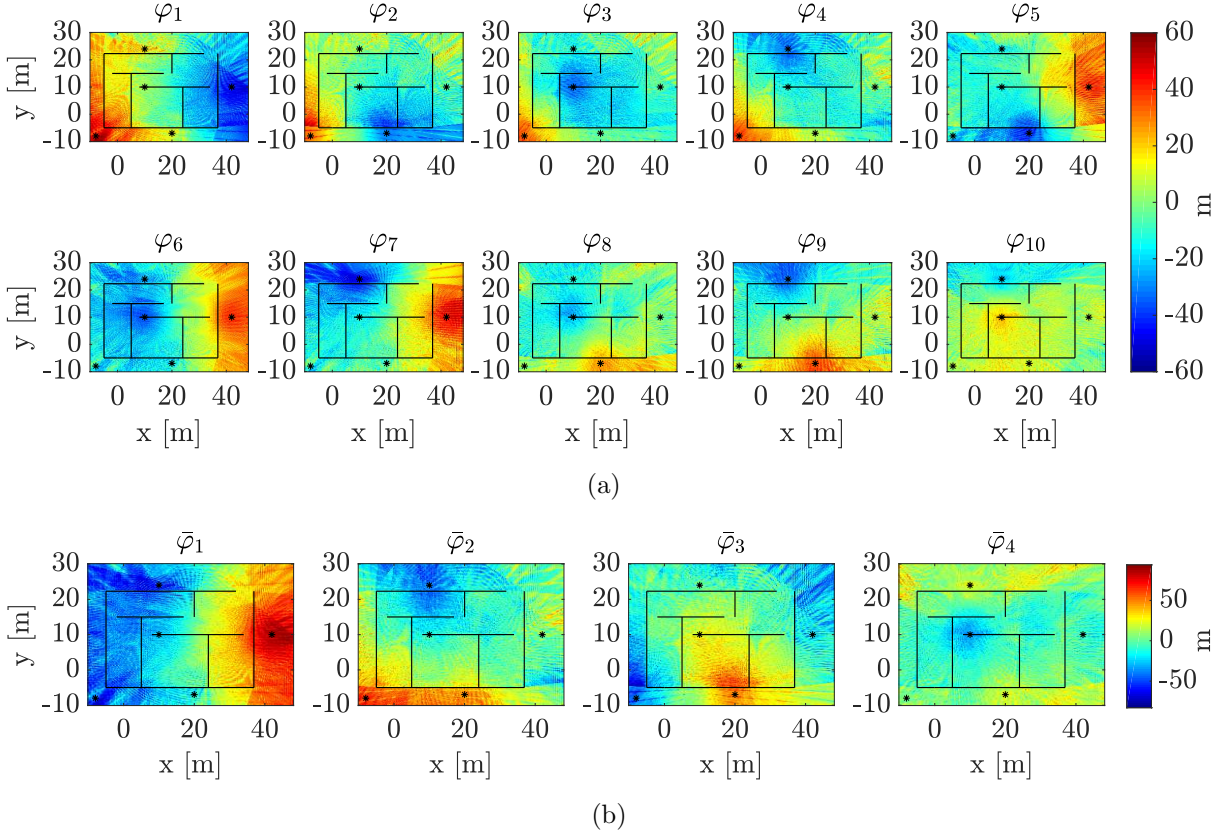


Figure B.10: Maps of: (a) all the  $M = 10$  features and (b)  $r = 4$  reduced features with  $L = 5$ ,  $B = 20$  MHz, and  $K = 10$ . The solid black lines represent the walls of the building and the black stars represent the transmitter locations. The maps are obtained by representing the value of the feature at every location in  $\mathcal{X}$ .

The reason is the redundancy inherent to TDoA features described in Sec. B.4.2. To ensure a fair comparison, LocB utilizes the same function learning algorithm as LocF; see Sec. B.3.2. Specifically, given  $\{(\hat{\mathbf{x}}_n, \tilde{p}_n)\}_{n=1}^N$ , the map is estimated as  $\hat{p}(\hat{\mathbf{x}}) = \boldsymbol{\kappa}'^\top(\hat{\mathbf{x}})\hat{\boldsymbol{\beta}}$  where  $\boldsymbol{\kappa}'(\hat{\mathbf{x}}) := [\kappa'(\hat{\mathbf{x}}, \hat{\mathbf{x}}_1), \dots, \kappa'(\hat{\mathbf{x}}, \hat{\mathbf{x}}_N)]^\top$ ,  $\hat{\boldsymbol{\beta}} := (\mathbf{K}' + \lambda' N \mathbf{I}_N)^{-1} \tilde{\mathbf{p}}$ , and  $\mathbf{K}'$  is an  $N \times N$  matrix with  $(n, n')$ -th entry  $\kappa'(\hat{\mathbf{x}}_n, \hat{\mathbf{x}}_{n'})$  and  $\kappa'$  is a Gaussian radial basis function with parameter  $\sigma'$ . In this way, this benchmark LocB algorithm coincides with those in [4, 13] when a power map must be estimated on a single frequency and with a single kernel. In all experiments, the values of  $\lambda$ ,  $\lambda'$ ,  $\sigma$ , and  $\sigma'$  used by the LocF and LocB schemes were tuned to approximately yield the lowest NMSE.

Fig. B.5 (left) depicts the true map generated through the multi-wall model, where the black crosses indicate the sensor locations and the solid white lines represent the walls of the building. The middle and right panels respectively show the LocB and LocF map estimates, obtained by placing a query sensor at every location. It is observed that the quality of the LocF estimate is considerably higher than that of the LocB estimate. The cause for the poor performance of the LocB algorithm is that the location estimates evolve in a non-smooth fashion across space, and attempting to learn the C2M from such non-smooth features is more challenging; see Figs. B.1c and B.1d and the discussion in Sec. B.1. To illustrate how the LocF approach alleviates this issue, Fig. B.6 depicts the



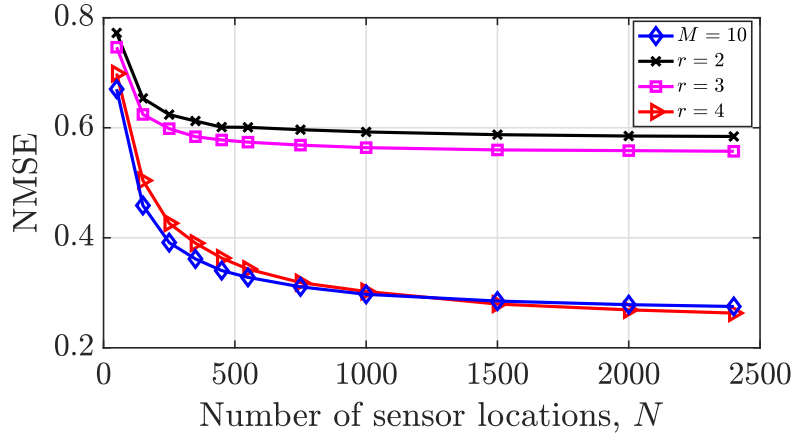


Figure B.11: Estimated map NMSE with reduced features for different  $r$  and without reduced features;  $L = 5$ ,  $B = 20$  MHz,  $K = 10$ ,  $\lambda = 1.6 \times 10^{-3}$ , and  $\sigma = 25$  m.

features used by the LocF estimator across  $\mathcal{X}$ . Specifically, if  $\phi(\mathbf{x})$  denotes the feature vector, obtained as in (B.28) for location  $\mathbf{x}$ , then the  $m$ -th panel titled  $\varphi_m$  in Fig. B.6 corresponds to the  $m$ -th entry of  $\phi(\mathbf{x})$  for each  $\mathbf{x} \in \mathcal{X}$ . It is observed that the evolution of these proposed features across space is significantly smoother than the one in Figs. B.1c and B.1d. A quantitative comparison is provided in Fig. B.7, which shows the NMSE as a function of the number of sensor locations  $N$  for  $L = 4$  and 7 transmitters. The error bars delimit intervals of 6 standard deviations of the NMSE across the 200 independent Monte Carlo runs. It is observed that, with high significance, the proposed LocF cartography scheme outperforms its LocB counterpart for both values of  $L$  provided that the number of measurement locations is roughly larger than 150.

The rest of the section studies the impact of multipath on the LocF and LocB cartography approaches by varying the number of walls. Fig. B.8 shows the NMSE as a function of the number of walls for different values of  $B$ . The parameters used for both LocF and LocB schemes are listed in Table B.1. The NMSE is obtained by also averaging over wall locations, which are confined to be in the positions of the walls in Fig. B.6 plus an additional wall that divides the room in two.

As expected, for all the simulated values of  $B$ , the performance of both LocF and LocB schemes is degraded (yet more severely in LocB) as the number of walls increases. Moreover, the performance of the LocB improves significantly with the bandwidth, since a higher bandwidth allows a more accurate estimation of the TDoA. This is because multipath components arriving within a time interval of length  $T = 1/B$  cannot be resolved; see Sec. B.4.1 and references therein. As intuition predicts, when multipath is sufficiently low and the bandwidth is sufficiently high, LocB cartography outperforms LocF. It is remarkable that LocF cartography exhibits robustness to multipath since the NMSE remains approximately constant even for a significant increase of multipath.

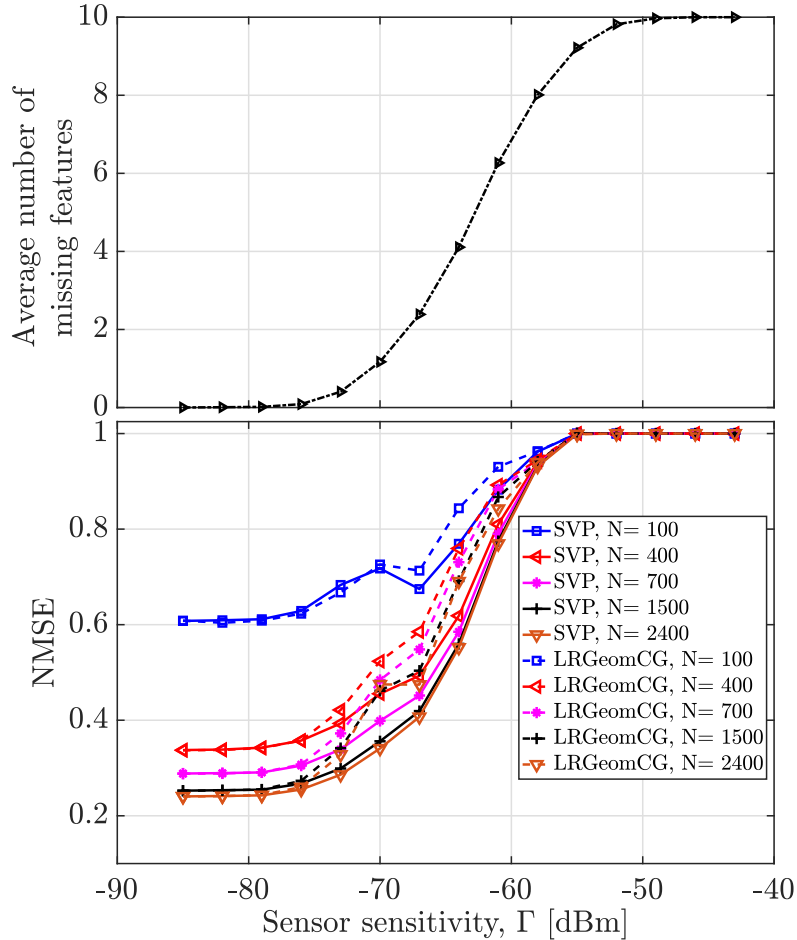


Figure B.12: (top) Average number of missing features and (bottom) estimated map NMSE, both as a function of  $\Gamma$  with  $L = 5$ ,  $B = 20$  MHz,  $K = 10$ ,  $\lambda = 1.9 \times 10^{-4}$ ,  $\mu = 5.42$ , and  $\sigma = 37$  m.

## B.5.2 Feature Design

This section provides empirical support for the findings in Sec. B.4.2. From now on, all experiments will involve only the LocF estimator. The first experiment investigates the impact of the number of features, which in all previous simulations was equal to  $M = L(L - 1)/2$ . To this end, Fig. B.9 shows the NMSE as a function of the number  $M$  of features for two different numbers  $N$  of sensor locations. The expectation operators in (B.27) also average with respect to all choices of  $M$  features out of the  $L(L - 1)/2$ . As observed, the NMSE improves from  $M = 4$  to roughly  $M = 7$  features, and remains approximately the same for  $M \geq 7$ . Although this effect may look counter-intuitive at first glance, this is a common phenomenon in machine learning related to the bias-variance trade-off [32] and the curse of dimensionality [31,32]; see Sec. B.3.1. Clearly, this effect motivates the feature dimensionality reduction techniques proposed in Sec. B.4.2. The rest of this section corroborates the merits of such techniques. A more challenging scenario with more walls will be considered. The first step is to determine the number of reduced features to be used. It can be seen that  $r = 4$  in (B.20) retains at least  $\eta = 99\%$  of the variance of the features in all tested scenarios. Thus, in principle, a

map can be learned using the reduced features  $\bar{\phi}_n := \mathbf{U}_1^\top \phi_n \in \mathbb{R}^4$  without meaningfully sacrificing estimation performance. Before corroborating that this is actually the case, it is instructive to visualize the aforementioned reduced features across space. Fig. B.10a portrays the maps of the  $M = 10$  original features, which correspond to the entries of  $\phi(\mathbf{x})$ ; see Sec. B.5.1. On the other hand, the panels of Fig. B.10b depict the reduced features over space, i.e., the 4 entries of the vector  $\bar{\phi}(\mathbf{x}) := \mathbf{U}_1^\top \phi(\mathbf{x})$  for each  $\mathbf{x} \in \mathcal{X}$ . These figures reveal that the reduced features inherit the spatial smoothness of the original features.

To quantify the impact of reducing the dimensionality of the feature vectors, Fig. B.11 compares the NMSE of the LocF map estimate that relies on the original features ( $M = 10$ ) with the one that relies on the reduced features ( $r = 2, 3, 4$ ). As observed, using just the 4 reduced features attains a similar performance to the estimator built on the 10 original features. This is expected given the bias-variance trade-off mentioned earlier. At this point, it might seem that the effects observed in Fig. B.9 contradict those of Fig. B.11 since in the former the NMSE is lower when 10 features are used relative to the case where only 4 are used. However, that should not be concluded since the features in Fig. B.9 correspond to the entries of  $\phi_n$  (see (B.28)) whereas the features in Fig. B.11 correspond to the entries of  $\bar{\phi}_n := \mathbf{U}_1^\top \phi_n$ .

### B.5.3 LocF cartography with Missing Features

This section assesses the performance of the techniques developed in Sec. B.4.3 to cope with missing features.

A feature will be deemed missing at a given sensor location if the received power of at least one of the two associated pilot signals is below a sensitivity threshold  $\Gamma$ . The top panel of Fig. B.12 depicts the average number of missing features as a function of  $\Gamma$ . The average is taken with respect to the sensor locations and noise. The bottom panel of Fig. B.12 shows the LocF map NMSE also as a function of  $\Gamma$ . The matrix completion problem in (B.23) is solved with both SVP and LRGeomCG; the implementation for the latter is the one provided in the ManOpt toolbox [49]. For higher values of  $N$ , the performance of both algorithms is clearly strongly determined by the average number of missing features. SVP seems to outperform LRGeomCG in terms of NMSE. Besides, the computation time of SVP is roughly half the one of LRGeomCG.

## B.6 Conclusions

Location-free (LocF) cartography has been proposed as an alternative to classical location-based (LocB) schemes, which suffer a strong performance degradation when multipath impairs the propagation of localization pilot signals. The central idea is to learn a map as a function of certain features of the localization pilot signals. Building upon this approach, kernel-ridge regression was applied to estimate power maps from these features. Practical issues addressed in the paper include feature design, dimensionality reduction, and dealing with missing features. Simulations corroborate the merits of LocF cartography relative to LocB alternatives. Future research will include mapping other channel metrics such



as power spectral density (PSD) and channel gain, as well as developing distributed and online extensions.

## References

- [1] A. Alaya-Feki, S. B. Jemaa, B. Sayrac, P. Houze, and E. Moulines, “Informed spectrum usage in cognitive radio networks: Interference cartography,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Cannes, France, Sep. 2008, pp. 1–5.
- [2] J.-A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.
- [3] B. A. Jayawickrama, E. Dutkiewicz, I. Oppermann, G. Fang, and J. Ding, “Improved performance of spectrum cartography based on compressive sensing in cognitive radio networks,” in *Proc. IEEE Int. Commun. Conf.*, Budapest, Hungary, Jun. 2013, pp. 5657–5661.
- [4] D. Romero, S.-J. Kim, G. B. Giannakis, and R. López-Valcarce, “Learning power spectrum maps from quantized power measurements,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2547–2560, May 2017.
- [5] S. Grimoud, S. B. Jemaa, B. Sayrac, and E. Moulines, “A REM enabled soft frequency reuse scheme,” in *Proc. IEEE Global Commun. Conf.*, Miami, FL, Dec. 2010, pp. 819–823.
- [6] E. Dall’Anese, S.-J. Kim, G. B. Giannakis, and S. Pupolin, “Power control for cognitive radio networks under channel uncertainty,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3541–3551, Aug. 2011.
- [7] W.C.M.V. Beers and J.P.C. Kleijnen, “Kriging interpolation in simulation: A survey,” in *Proc. IEEE Winter Simulation Conf.*, Washington, D. C., Dec. 2004, vol. 1, pp. 113–121.
- [8] G. Boccolini, G. Hernandez-Penalosa, and B. Beferull-Lozano, “Wireless sensor network for spectrum cartography based on kriging interpolation,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Sydney, NSW, Nov. 2012, pp. 1565–1570.
- [9] G. Ding, J. Wang, Q. Wu, Y.-D. Yao, F. Song, and T. A. Tsiftsis, “Cellular-base-station-assisted device-to-device communications in TV white space,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 107–121, Jul. 2016.
- [10] S.-J. Kim, N. Jain, G. B. Giannakis, and P. Forero, “Joint link learning and cognitive radio sensing,” in *Proc. Asilomar Conf. Signal, Syst., Comput.*, Pacific Grove, CA, Nov. 2011, pp. 1415–1419.

- [11] S.-J. Kim and G. B. Giannakis, “Cognitive radio spectrum prediction using dictionary learning,” in *Proc. IEEE Global Commun. Conf.*, Atlanta, GA, Dec. 2013, pp. 3206–3211.
- [12] D.-H. Huang, S.-H. Wu, W.-R. Wu, and P.-H. Wang, “Cooperative radio source positioning and power map reconstruction: A sparse Bayesian learning approach,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2318–2332, Aug. 2014.
- [13] J.-A. Bazerque and G. B. Giannakis, “Nonparametric basis pursuit via kernel-based learning,” *IEEE Signal Process. Mag.*, vol. 28, no. 30, pp. 112–125, Jul. 2013.
- [14] M. Hamid and B. Beferull-Lozano, “Non-parametric spectrum cartography using adaptive radial basis functions,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, New Orleans, LA, Mar. 2017, pp. 3599–3603.
- [15] J.-A. Bazerque, G. Mateos, and G. B. Giannakis, “Group-lasso on splines for spectrum cartography,” *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4648–4663, Oct. 2011.
- [16] S.-J. Kim, E. Dall’Anese, and G. B. Giannakis, “Cooperative spectrum sensing for cognitive radios using Krige Kalman filtering,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, Jun. 2010.
- [17] D. Romero, D. Lee, and G. B. Giannakis, “Blind channel gain cartography,” in *Proc. IEEE Global Conf. Signal Inf. Process.*, Greater Washington, D. C., Dec. 2016, pp. 1110–1115.
- [18] D. Lee, S.-J. Kim, and G. B. Giannakis, “Channel gain cartography for cognitive radios leveraging low rank and sparsity,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5953–5966, Jun. 2017.
- [19] D. Lee, D. Berberidis, and G. B. Giannakis, “Adaptive Bayesian channel gain cartography,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Calgary, Canada, Apr. 2018, pp. 3555–3558.
- [20] M. Bshara, U. Orguner, F. Gustafsson, and L. Van Biesen, “Fingerprinting localization in wireless networks based on received-signal-strength measurements: A case study on WiMAX networks,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 283–294, Aug. 2009.
- [21] P. S. Naidu, *Distributed Sensor Arrays: Localization*, CRC Press, 2017.
- [22] A. Bensky, *Wireless Positioning Technologies and Applications*, Artech House, 2016.
- [23] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” *IEEE Trans. Syst., Man, Cybernetics C., Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

- [24] InfSoft, “Ultra-wideband,” [Online]. Available: <https://www.ultrawideband.io/en/technology.php>.
- [25] L. Yang and G. B. Giannakis, “Ultra-wideband communications: An idea whose time has come,” *IEEE Signal Process. Mag.*, vol. 21, no. 6, pp. 26–54, Nov. 2004.
- [26] D. Dardari, C.-C. Chong, and M. Win, “Threshold-based time-of-arrival estimators in UWB dense multipath channels,” *IEEE Trans. Commun.*, vol. 56, no. 8, pp. 1366–1378, Aug. 2008.
- [27] M. Brunato and R. Battiti, “Statistical learning theory for location fingerprinting in wireless LANs,” *Computer Networks*, vol. 47, no. 6, pp. 825–845, Apr. 2005.
- [28] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, “On indoor position location with wireless LANs,” in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Lisboa, Portugal, Sep. 2002, vol. 2, pp. 720–724.
- [29] HP, “SmartLOCUS,” [Online]. Available: <https://www.rfidjournal.com>.
- [30] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “LANDMARC: Indoor location sensing using active RFID,” in *Proc. IEEE Int. Conf. Pervasive Computing Commun.*, Fort Worth, TX, Mar. 2003, pp. 407–415.
- [31] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics. Springer, 2006.
- [32] V. Cherkassky and F. M. Mulier, *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, 2007.
- [33] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.
- [34] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in *Proc. Comput. Learning Theory*, Amsterdam, The Netherlands, Jul. 2001, pp. 416–426.
- [35] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.
- [36] S. W. Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing*, California Technical Publishing, 1997.
- [37] N. E. Gemayel, S. Koslowski, F. K. Jondral, and J. Tschan, “A low cost TDOA localization system: Setup, challenges and results,” in *Proc. Workshop Pos. Navigation Commun.*, Dresden, Germany, Mar. 2013, pp. 1–4.
- [38] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Stanford University, 2002.
- [39] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Foundations Comput. Math.*, vol. 9, no. 6, pp. 717–772, Apr. 2009.

- [40] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM J. Opt.*, vol. 52, no. 3, pp. 471–501, Aug. 2010.
- [41] E. J. Candès and T. Tao, “The power of convex relaxation: Near-optimal matrix completion,” *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, Apr. 2010.
- [42] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2009.
- [43] B. Vandereycken, “Low-rank matrix completion by riemannian optimization,” *SIAM J. Opt.*, vol. 23, no. 2, pp. 1214–1236, Jun. 2013.
- [44] P. Jain, R. Meka, and I. S. Dhillon, “Guaranteed rank minimization via singular value projection,” in *Proc. Advances in Neural Inf. Proces. Syst.*, Vancouver, Canada, Dec. 2010, pp. 937–945.
- [45] S. Hosseinzadeh, H. Larijani, and K. Curtis, “An enhanced modified multi wall propagation model,” in *Proc. IEEE Global Internet of Things Summit*, Geneva, Switzerland, Jun. 2017, pp. 1–4.
- [46] C. A. Micchelli, Y. Xu, and H. Zhang, “Universal kernels,” *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, Dec. 2006.
- [47] D. Ismailova and W.-S. Lu, “Improved least-squares methods for source localization: An iterative re-weighting approach,” in *Proc. IEEE Int. Conf. Dig. Signal Process.*, Singapore, Jul. 2015, pp. 665–669.
- [48] R. Kaune, “Accuracy studies for TDOA and TOA localization,” in *Proc. IEEE Int. Conf. Inf. Fusion*, Singapore, Jul. 2012, pp. 408–415.
- [49] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, “ManOpt, a MATLAB toolbox for optimization on manifolds,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1455–1459, Apr. 2014.