

DISTRIBUTED ON-LINE MULTIDIMENSIONAL SCALING FOR SELF-LOCALIZATION IN WIRELESS SENSOR NETWORKS

Gemma Morral[†], Ndeye Amy Dieng[†] and Pascal Bianchi[†]

[†]Institut Mines-Télécom, Télécom ParisTech, CNRS LTCI

46, rue Barrault, 75634 Paris Cedex 13, France

Email: morralad,pascal.bianchi,ndeye.dieng@telecom-paristech.fr

ABSTRACT

Consider a wireless network formed by fixed or mobile nodes. Each node seeks to estimate its own position based on noisy measurements of the relative distance with other nodes. In a centralized batch mode, positions can be retrieved (up to a rigid transformation) by applying Principal Component Analysis (PCA) on a so-called similarity matrix built from the relative distances. In this paper, we propose a distributed on-line algorithm allowing each node to estimate its own position based limited exchange of information in the network. Our framework encompasses the case of sporadic measurements and random link failures. We prove the consistency of our algorithm in the case of fixed sensors. Our numerical results also demonstrate the attractive performance of the algorithm for tracking the positions of mobile sensors. Simulations are conducted on a wireless sensor network testbed.

Index Terms— Distributed algorithms, On-line algorithms, Localization, Principal Component Analysis, Wireless Sensor Networks.

1. INTRODUCTION

Self-localization in Wireless Sensor Networks (WSN) have raised a great deal of attention in the signal processing community during the last decades (see [1] or [2] for an overview). In many situations, sensors are unaware of their true positions, but are able to measure their distance with other neighboring sensors. For instance, the inter-sensor distances can be estimated by means of the Received Signal Strength (RSS) intensity, which is directly related to the distance. In such situation, the aim is to build a map of the sensors' positions based on the relative distances. In the literature, the task is often made easier by the assumption that the network contains a significant number of *anchor-nodes* whose position is perfectly known. In other situations, specially in distributed mobile sensor networks, such anchor nodes may not be present, or too far away. One should therefore rely on *anchor-free* method.

The problem is often referred to as multidimensional scaling (MDS). The MDS-MAP algorithm is based on the analysis of the principal components (PCA) of a so-called similarity matrix constructed from the relative sensors distances (see [3], [4] for a complete description and analysis). Alternative approaches for MDS on the localization context include : stress function majorization (see [5] Chapter 8), the so-called local-MDS-MAP algorithm [6] which relies on merging local maps and several optimization techniques using semidefinite programming (see [7, 8]), a mass-spring model as in [9] or kernel regression methods as in [10].

Standard algorithms such as those mentioned above are *batch* algorithms in the sense that the positions are estimated only once – after the prior estimation of relative distances. In this paper, we focus on *on-line* approaches: each node makes use of a new observation to update the current estimate of its position, and delete its observation after use. This context is specially useful in the case when the sensors need to keep track of their position. In that case, an online version of the MDS-MAP algorithm can be obtained by using the so-called Oja's algorithm [11, 12] which extracts a sought principal eigenspace.

More recently, the emergence of distributed sensor network raised the question of implementing such algorithms in a distributed fashion [13]. A distributed (batch) algorithm for stress function majorization based on a round-robin communication scheme has been proposed in [14]. On-line distributed gossip-based algorithms have been proposed in [15] and [16]. A distributed version of [9] can be found in [17]. More recently, several attempts have been made to extend Oja's algorithm to a distributed asynchronous setting [18] [19] but the applicability of such approaches to the problem of self-localization remains unexplored.

In this paper, we propose a distributed online anchor-free method for self-localization. Our algorithm is asynchronous and encompasses the case of random link failures and random noisy and sporadic RSS measurements. The paper is organized as follows. Section 2 introduces the framework and recalls the standard centralized batch MDS approach for localization. Section 3 investigates the case of centralized on-

The work of G. Morral is supported by DGA (French Armement Procurement Agency) and the Institut Mines-Telecom.

line MDS. Section 4 introduces our algorithm and states the main convergence result. Numerical results on a testbed are provided in Section 5.

2. THE FRAMEWORK

2.1. Context and objective

Consider N agents seeking to estimate their respective positions $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$. We assume that \mathbf{x}_i are row-vectors in $\mathbb{R}^{1 \times p}$ for any i , where $p < N$ (in practice, $p = 2$ or 3). We assume that agents have access to noisy measurements of relative distances $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ for all $i, j \in \{1, \dots, N\}$. Based on these noisy measurements, the objective of the network is to estimate the agent's positions each time new observations are made.

It is worth noting that the above problem is in fact ill-posed. Exact positions are identifiable only up to a rigid transformation. Indeed, quantities $(d_{i,j})_{i,j}$ are preserved when an isometry is applied to the agents' positions. The problem is generally circumvented by assuming that some of the agents (called anchor nodes) are perfectly aware of their position.

2.2. Centralized batch MDS

Although this paper focuses on online localization, it is useful for the sake of completeness to first mention the standard *batch* MDS approach. Define S as the $N \times N$ matrix of square relative distances *i.e.*, $S(i, j) = d_{i,j}^2$. Define $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ as the center of mass of the agents. Upon noting that $d_{i,j}^2 = \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 + \|\mathbf{x}_j - \bar{\mathbf{x}}\|^2 - 2\langle \mathbf{x}_i - \bar{\mathbf{x}}, \mathbf{x}_j - \bar{\mathbf{x}} \rangle$, one has:

$$S = \mathbf{c}\mathbf{1}^T + \mathbf{1}\mathbf{c}^T - 2XX^T \quad (1)$$

where T stands for transpose, $\mathbf{1}$ is the $N \times p$ matrix whose components are all equal to one, $\mathbf{c} = (\|\mathbf{x}_1 - \bar{\mathbf{x}}\|^2, \dots, \|\mathbf{x}_N - \bar{\mathbf{x}}\|^2)^T$ and X is a $N \times p$ matrix whose i th line coincides with the row-vector $\mathbf{x}_i - \bar{\mathbf{x}}$. Otherwise stated, the i th line of X coincides with the *barycentric coordinates* of agent i .

Define $J = \mathbf{1}\mathbf{1}^T/N$ as the orthogonal projector onto the linear span of $\mathbf{1}$. Define $J_\perp = I_N - J$ as the projector onto the space of vectors with zero sum, where I_N is the $N \times N$ identity matrix. It is straightforward to verify that $J_\perp X = X$. Thus, introducing the matrix

$$M \triangleq -\frac{1}{2}J_\perp S J_\perp,$$

equation (1) implies that $M = XX^T$. In particular, M is symmetric non-negative and has rank (at most) p . The agents' coordinates can be recovered from M (up to a rigid transformation) by recovering the principal eigenspace of M *i.e.*, the vector-space spanned by the p th principal eigenvectors (see Chapter 12 in [5]). Denote by $\{\sigma_k\}_{k=1}^N$ the eigenvalues of M in decreasing order. In the sequel, we shall always assume

that $\sigma_p > 0$, meaning that M has a full column-rank p . Denote by $\{\mathbf{u}_k\}_{k=1}^p$ corresponding unit-norm $N \times 1$ eigenvectors. Set $\hat{X} = (\sqrt{\sigma_1}\mathbf{u}_1, \dots, \sqrt{\sigma_p}\mathbf{u}_p)$. Clearly $M = \hat{X}\hat{X}^T$ and $X = Q\hat{X}$ for some matrix Q such that $QQ^T = I_N$. Otherwise stated, \hat{X} coincides with the barycentric coordinates X up to an orthogonal transformation. In practice, matrix S is usually not perfectly known and must be replaced by an estimate. This yields **Algorithm 1** (Chapter 12 in [5]).

Algorithm 1 Centralized batch MDS for localization

Input: Noisy estimates $\hat{d}_{i,j}$ of $d_{i,j}$ for all i, j .

1. Compute matrix $\hat{S} = (\hat{d}_{i,j}^2)_{i,j=1,\dots,N}$.

2. Set $\hat{M} = -\frac{1}{2}J_\perp \hat{S} J_\perp$.

3. Find the p principal eigenvectors $\{\mathbf{u}_k\}_{k=1}^p$ and eigenvalues $\{\sigma_k\}_{k=1}^p$ of \hat{M} .

Output: $\hat{X} = (\sqrt{\sigma_1}\mathbf{u}_1, \dots, \sqrt{\sigma_p}\mathbf{u}_p)$

3. CENTRALIZED ONLINE MDS

From now on, we focus on *online* localization.

3.1. Observation Model

At each time instant n , we assume that with probability $q_{i,j}$, an agent i is able to obtain an estimate $S_n(i, j)$ of the square distance with another agent $j \neq i$. The most typical case is when Received Signal Strength (RSS) measurements are made. A traditional model in wireless sensor network [20] assumes that the average attenuation $P_{i,j}$ between nodes i and j is related to the distance $d_{i,j}$ through $P_{i,j} = P_0 - 10\eta \log_{10} d_{i,j}/d_0$. Parameters P_0 , η and d_0 are predetermined constants which depend on the transmission medium. We introduce a collection of independent random variables $(P_n(i, j) : i, j = 1, \dots, N, n \in \mathbb{N})$ such that $P_n(i, j)$ follows a Gaussian distribution of mean $P_{i,j}$ and variance σ^2 . It is easy to verify from [21] that the quantity

$$D_n(i, j) \triangleq d_0^2 10^{\frac{P_n(i, j) - P_0}{5\eta}} 10^{-\frac{\sigma^2 \ln 10}{50\eta^2}} \quad (2)$$

is an unbiased estimate of $d_{i,j}^2$ *i.e.*, $\mathbb{E}(D_n(i, j)) = d_{i,j}^2$. We set $D_n(i, i) = 0$. We assume that node i makes an observation and computes the estimate $D_n(i, j)$ with probability $q_{i,j}$ and makes no observation otherwise. Thus, one can represent the available observations as the product $B_n(i, j)D_n(i, j)$ where $(B_n)_n$ is an i.i.d. sequence of random matrices whose components $B_n(i, j)$ follow the Bernoulli distribution of parameter $q_{i,j}$. Stated otherwise, node i observes the i th line of matrix $B_n \circ D_n$ at time n where \circ stands for the Hadamard product.

Lemma 1. Assume $q_{i,j} > 0$ for all pairs i, j . Set $W := [q_{i,j}^{-1}]_{i,j=1}^N$ and let B_n, S_n be defined as above. The matrix

$$S_n \triangleq W \circ B_n \circ D_n \quad (3)$$

is an unbiased estimate of S i.e., $\mathbb{E}(S_n) = S$.

3.2. Oja's online PCA

Our objective is to eventually find a point in the set χ of $N \times p$ matrices whose columns are orthonormal and span the vector space associated with the p principal eigenvalues of M .

As a consequence of Lemma 1, an unbiased estimate of M is simply obtained by $M_n = -\frac{1}{2}J_\perp S_n J_\perp$. When faced with random matrices M_n having a given expectation M , the principal eigenspace of M can be recovered by the following algorithm, due to Oja [22] and analyzed in [23]. The algorithm generates a sequence $(U_n)_n$ of $N \times p$ matrices according to:

$$U_n = U_{n-1} + \gamma_n \left(M_n U_{n-1} - U_{n-1} (U_{n-1}^T M_n U_{n-1}) \right), \quad (4)$$

where $\gamma_n > 0$ is a step size.

In order to have more insight, it is convenient to interpret (4) as a Robbins-Monro algorithm (see Chapter 3, [24]) of the form $U_n = U_{n-1} + \gamma_n (h(U_{n-1}) + \xi_n)$ where ξ_n is a martingale increment noise and h is the so-called *mean field* of the algorithm given by $h(U) = MU - UU^T MU$. It is known that under adequate stability assumptions and vanishing step size γ_n , the algorithm converges to the roots of h (Theorem 2 in [24]). By Theorem 1 of [11] the roots of h are essentially rotations of matrices whose columns are eigenvectors of M , multiplied by some scalar, including zero. Thus, strictly speaking, the algorithm might converge to a broader set than the sought set χ . Fortunately, it is known since [25] that all roots of h outside the set χ are unstable. Undesired points can be avoided by standard *avoidance-of-traps* methods (see Chapter 4 in [26] and [27]).

In practice, the algorithm (4) is known to suffer from numerical instabilities depending on the initialization [23]. However, since we are expecting convergence to unit-norm vector, these instabilities can be avoided by introducing a projection step:

$$U_n = \Pi_K \left[U_{n-1} + \gamma_n \left(M_n U_{n-1} - U_{n-1} (U_{n-1}^T M_n U_{n-1}) \right) \right], \quad (5)$$

where K is any arbitrary compact convex set whose interior contains χ , and where Π_K is the projector onto K . For completeness, we mention that the authors of [12] proposed an alternative normalization procedure which allows to avoid projection, but which seems difficult to generalize to the distributed context.

3.3. Localization

Let $\mathbf{u}_{n,k}$ denote the k th column of matrix U_n . If $(\mathbf{u}_{n,k})_n$ converges to one of the eigenvectors of M , then the quantity $\sigma_{n,k}$ recursively defined by $\sigma_{n,k} = (1 - \gamma_n)\sigma_{n-1,k} + \gamma_n \mathbf{u}_{n-1,k}^T M_n \mathbf{u}_{n-1,k}$ converges to the corresponding eigenvalue (see [23]). Finally, an estimate of the barycentric coordinates are obtained by $\hat{X}_n = \text{diag}(\sqrt{\sigma_{n,1}}, \dots, \sqrt{\sigma_{n,p}})U_n$.

4. DISTRIBUTED ONLINE MDS

4.1. Sparse Asynchronous Communications

It is clear from the previous section that an unbiased estimate of matrix M is the first step needed to estimate the sought eigenspace. In the centralized setting, this estimate was given by matrix $M_n = -\frac{1}{2}J_\perp S_n J_\perp$. As made clear in Section 3.1, **each node i observes the i th row of matrix S_n** . As a consequence, node i has access to the i th row-average $\bar{S}_n(i) \triangleq \frac{1}{N} \sum_j S_n(i,j)$. This means that matrix $S_n J_\perp$ can be obtained with no need to further exchange of information in the network. On the other hand, $J_\perp S_n J_\perp$ requires to compute the per-column averages of matrix $S_n J_\perp$. This task is difficult in a distributed setting, as it would require that all nodes share all their observations at any time. A similar obstacle happens in Oja's algorithm when computing matrix products. To circumvent the above difficulties, we introduce the following sparse asynchronous communication framework.

At time n , we assume that a given node ι_n wakes up and transmits a message to other nodes. A given node $i \neq \iota_n$ receives the latter message with probability q .

Definition 1 (Asynchronous Transmission Sequence). *Let q be a real number such that $0 < q < 1$. We say that the sequence of random vectors $T_n = (\iota_n, Z_{n,i} : i \in \{1, \dots, N\}, n \in \mathbb{N})$ is an Asynchronous Transmission Sequence (ATS) if: i) all variables $(\iota_n, Z_{n,i})_{i,n}$ are independent, ii) ι_n is uniformly distributed on the set $\{1, \dots, N\}$, iii) $\forall i \neq \iota_n, Z_{n,i}$ is a Bernoulli variable with parameter q i.e., $\mathbb{P}[Z_{n,i} = 1] = q$ and iv) $Z_{n,\iota_n} = 0$.*

4.2. The Algorithm

Similarly to (4), our aim is to iteratively generate a sequence of $N \times p$ matrices U_n . In our distributed framework, **each node i is in charge of the update of the i th row of U_n** denoted by $U_n(i)$ (not to be confused with the k th column previously denoted by $\mathbf{u}_{n,k}$).

Consider a ATS $(T_n)_n$. Assume that the active node ι_n broadcasts its former estimates $U_{n-1}(\iota_n)$ and $\bar{S}_n(i)$. All nodes i such that $Z_{n,i} = 1$ receive the estimates. Thus, all nodes compute:

$$Y_n(i) = \widehat{M}_n(i, i)U_{n-1}(i) + \frac{N}{q}U_{n-1}(\iota_n)\widehat{M}_n(i, \iota_n)Z_{n,i} \quad (6)$$

where for any i, j ,

$$\widehat{M}_n(i, j) = \frac{\bar{S}_n(i) + \bar{S}_n(j)}{2} - \frac{S_n(i, j) + \delta_n(i)}{2} \quad (7)$$

$$\delta_n(i) = \frac{\bar{S}_n(i)}{N} + \frac{\bar{S}_n(\iota_n)Z_{n,i}}{q}. \quad (8)$$

As will be made clear below, the matrix Y_n whose i th row coincides with $Y_n(i)$ can be interpreted as an unbiased estimate of MU_{n-1} i.e., $\mathbb{E}(Y_n|U_{n-1}) = MU_{n-1}$.

Consider a second ATS $(T'_n)_n$. At time n , node t'_n wakes up and broadcasts the product $U_{n-1}(t'_n)^T Y_n(t'_n)$ to other nodes. Receiving nodes are those i 's for which $Z'_{n,i} = 1$. Similarly to (5), we set:

$$U_n(i) = \Pi_I [U_{n-1}(i) + \gamma_n (Y_n(i) - U_{n-1}(i)\Sigma_n(i))] , \quad (9)$$

$$\Sigma_n(i) = U_{n-1}(i)^T Y_n(i) + \frac{N}{q} U_{n-1}(t'_n)^T Y_n(t'_n) Z'_{n,i} . \quad (10)$$

where for any i Π_I is the projector onto the set $I \triangleq [-r, r]^p$ for an arbitrary $r > 1$ and $\Sigma_n(i)$ is a $p \times p$ matrix.

To obtain the estimate position $\hat{X}_n(i)$ at each sensor i as in Section 3.3, the sequence of p eigenvalues are generated at each i as the following square matrix $p \times p$:

$$\begin{aligned} \sigma_n(i) &= \sigma_{n-1}(i) + \gamma_n (\text{diag}(\Sigma_n(i)) - \sigma_{n-1}(i)) \\ \text{then, } \hat{X}_n(i) &= U_n(i) \sqrt{\sigma_n(i)} \end{aligned} \quad (11)$$

Algorithm 2 Distributed On-line MDS for localization

At each time $n = 1, 2, \dots$

[Local step]:

- Nodes make sparse measurements of their respective RSS.
- Each node i evaluates $(S_n(i, j), j = 1, \dots, N)$ and $\bar{S}_n(i)$ using (2) and (3).

[Communication step]:

- A node t_n randomly selected broadcasts $U_{n-1}(t_n)$ and $\bar{S}_n(t_n)$ to nodes i such that $Z_{n,i} = 1$.
 - Each node i computes $Y_n(i)$ by (6).
 - A node t'_n randomly selected broadcasts $U_{n-1}(t'_n)^T Y_n(t'_n)$ to nodes i such that $Z'_{n,i} = 1$.
 - Each node i updates $U_n(i)$ and $\hat{X}_n(i)$ by (9) and (10)-(11).
-

4.3. Convergence analysis

Regarding the convergence, we prove that if the sensors' positions are fixed, the algorithm recovers the latter up to a rigid transformation.

Assumption 1 (Step size). *Sequence $(\gamma_n)_n$ satisfies $\gamma_n > 0 \rightarrow 0$, $\sum_n \gamma_n = +\infty$ and $\sum_n \gamma_n^2 < \infty$.*

Proposition 1. *For any $U \in \mathbb{R}^{N \times p}$, set $h(U) = MU - UU^T MU$. Let U_n be defined by (9). There exists two random sequences $(\xi_n, e_n)_n$ such that, almost surely (a.s.), e_n converges to zero, $\sum_n \gamma_n \xi_n$ converges and*

$$U_n = U_{n-1} + \gamma_n (h(U_{n-1}) + \xi_n + e_n) . \quad (12)$$

Theorem 1 (Main result). *Let U_n be defined by (9) and $\sigma_{n,k}$ be defined by (11). Under Assumption 1, for any $k = 1, \dots, p$, the k th column $\mathbf{u}_{n,k}$ of U_n converges to an eigenvector of M with unit-norm. Moreover, $\sigma_{n,k}$ converges to the corresponding eigenvalue.*

Proof. The proof is an immediate consequence of Proposition 1 along with Theorem 2 of [24]. Sequence U_n converges a.s. to the roots of h . The latter roots are characterized in [11]. In particular, $h(U) = 0$ implies that each column of U is a unit-norm eigenvector of M . The detailed proofs will be given within an extension version of this work. ■

Note that Theorem 1 might seem incomplete in some respect: one indeed expects that the sequence U_n converges to the set χ characterizing the principal eigenspace of M . Instead, Theorem 1 only guarantees that one recovers *some* eigenspace of M . As discussed in Section 3.2, undesired limit points can be theoretically avoided by introducing an arbitrary small Gaussian noise inside the parenthesis of the lefthand side of (9) (see Chapter 4 in [26]).

5. NUMERICAL RESULTS

We consider a set of $N = 10$ nodes selected from the SensLAB platform [28] (CC2420 sensor nodes which can be managed by uploaded firmwares) which are able to exchange packets and collect corresponding RSS measures. The estimated parameters are : $\sigma^2 = 16.4$, $P_0 = -60.3$ and $\eta = 2.1$. As described in Section 4 we set $q_{ij} = 0.8 \forall i, j$, $q = 0.5$ and $\gamma_t = \frac{0.015}{\sqrt{t}}$. Figure 1 shows the average estimated relative positions over 100 independent runs after 2000 iterations of Algorithm 2.

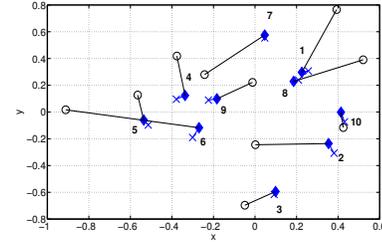


Fig. 1: Denote as : (O) the random initialized positions, (◆) the real relative positions and (X) the average estimated.

To illustrate the impact of the level noise σ^2 and communication parameters q_{ij} and q , we draw a graph of $N = 10$ randomly located within the unit square $[0, 1] \times [0, 1]$. Figure 2 displays a comparison about the mean error of both first and second estimated eigenvectors for different sparsity degrees (q_{ij} and q) and considering both noiseless and noisy cases.

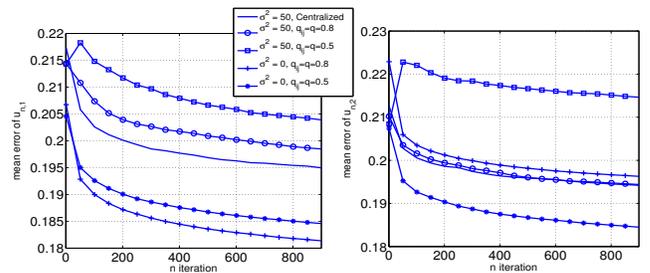


Fig. 2: Root mean square error as a function of the number of iterations n from the first estimated eigenvector $\mathbf{u}_{n,1}$ (on the right) and the second estimated eigenvector $\mathbf{u}_{n,2}$ (on the left).

6. REFERENCES

- [1] Mao G., Fidan B., and Anderson B.D.O. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, 2007.
- [2] Kotwal S. B., Verma S., and Abrol R. K. Approaches of Self Localization in Wireless Sensor Networks and Directions in 3D. *International Journal of Computer Applications*, 50(11):1–10, 2012.
- [3] Shang Y., Ruml W., and Fromherz M. Localization from Mere Connectivity. In *MobiHoc*, Anapolis, 2003.
- [4] Oh S., Montanari A., and Karbasi A. Sensor Network Localization from Local Connectivity : Performance Analysis for the MDS-MAP Algorithm. In *ITW*, Cairo, 2010.
- [5] Borg I. and Groenen P. *Modern Multidimensional Scaling: theory and applications*. New York: Springer-Verlag, 2005.
- [6] Shang Y. and Ruml W. Improved MDS-based localization. In *INFOCOM*, Hong Kong, 2004.
- [7] Chen H., Wang G., Wang Z., So H., and Poor H.V. Non-Line-of-Sight Node Localization Based on Semi-Definite Programming in Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 11(1):108–116, 2012.
- [8] Javanmard A. and Montanari A. Localization from Incomplete Noisy Distance Measurements. *Journal of Foundations of Computational Mathematics*, 13(3):297–345, 2013.
- [9] Priyantha N.B., Hari B., Erik D., and Seth T. Anchor-Free Distributed Localization in Sensor Networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 340–341. ACM, 2003.
- [10] Honeine P., Richard C., Bermudez J.C.M., Snoussi H., and et al. Functional estimation in Hilbert space for distributed learning in wireless sensor networks. In *ICASSP*, Taipei, 2009.
- [11] Oja E. Principal components, minor components, and linear neural networks. *Journal of Neural Networks*, 5(6):927–935, 1992.
- [12] Borkar V. and Meyn S.P. Oja's algorithm for graph clustering, markov spectral decomposition, and risk sensitive control. *Journal of Automatica*, 48(10):2512–2519, 2012.
- [13] Patwari N. and et al. Locating the Nodes : cooperative localization in wireless sensor networks]. *IEEE Signal Processing Magazine*, 22(4):54–69, 2005.
- [14] Costa J. A., Patwari N., and O.Hero A. Distributed Weighted-Multidimensional Scaling for Node Localization in Sensor Networks. *ACM Transactions on Sensor Networks*, 2(1):39–64, 2006.
- [15] Cattivelli F.S. and Sayed A.H. Distributed nonlinear Kalman filtering with applications to wireless localization. In *ICASSP*, Dallas, 2010.
- [16] Trawny N., Roumeliotis S.I., and Giannakis G.B. Cooperative multi-robot localization under communication constraints. In *ICRA*, Kobe, 2009.
- [17] Yu J., Kulkarni S. R., and Poor H.V. Dimension expansion and customized spring potentials for sensor localization. *EURASIP Journal on Advances in Signal Processing*, 2013:20, 2013.
- [18] Morral G., Bianchi P., and Jakubowicz J. Asynchronous Distributed Principal Component Analysis Using Stochastic Approximation. In *CDC*, Hawaii, 2012.
- [19] Tomozei D.C. and Massoulié L. Distributed user profiling via spectral methods. In *SIGMETRICS*, New York, 2010.
- [20] Rappaport T. S. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.
- [21] Zang C., Shenyang W.L., and Yu H. The Probabilistic Analysis of Distance Estimators in Wireless Sensor Network. In *ICNC*, Haikou, 2007.
- [22] Oja E. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.
- [23] Oja E. and Karhunen J. On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix. *Journal of Mathematical Analysis and Applications*, 106(1):69–84, 1985.
- [24] Delyon B. Stochastic Approximation with Decreasing Gain: Convergence and Asymptotic Theory. *Unpublished Lecture Notes*, http://perso.univ-rennes1.fr/bernard.delyon/as_cours.ps, 2000.
- [25] Williams R. Feature discovery through error-correcting learning. Technical report, CA : University of California, Institute of Cognitive Science, 1985.
- [26] Borkar V.S. *Stochastic approximation: a dynamical system viewpoint*. Cambridge University Press, 2008.
- [27] Pelletier M. Weak convergence rates for stochastic approximation with application to multiple targets and simulated annealing. *Annals of Applied Probability*, 8(1):10–44, 1998.
- [28] SensLAB Platform Rennes. <https://www.senslab.info/>.