# Self-Synchronization Based Localization of a Time-Misaligned Transmitter in Cellular Networks

Evert Pocoma, ULB and François Quitin, ULB and Luc Vandendorpe, UCL and Philippe De Doncker, ULB and François Horlin, ULB

Abstract—Geo-localization services are an important functionality in cellular networks. Besides, the use of Ultra Dense Networks and small cells, in current and future cellular networks, greatly increases the complexity of centralized localization approaches. Consequently, we propose a Self-Synchronization Positioning Estimation (SSPE) algorithm that estimates the transmitter position in a distributed fashion. The proposed SSPE algorithm reaches consensus for the posterior distribution of the transmitter position rather than on the final estimates. Such consensus ensures that the proposed SSPE algorithm converges to the centralized Direct Positioning Estimation (DPE) approach, which has the best performance of all localization approaches. We show that the proposed algorithm is related to the Iterative Positioning Estimation (IPE) algorithm, since both exploit the self-synchronization mechanism. As a result, the improvements and extensions for IPE, previously studied in other works, can be directly applied to the proposed SSPE algorithm. In addition, the proposed algorithm is able to localize the transmitter even when it is not time synchronized with the network as it is usually the case. The performance of the algorithms is numerically assessed through Monte-Carlo simulations by the mean distance error and mean range offset error. Finally, we not only show that our approach gets close to the DPE performance after a few iterations, but also that it converges for different logical network configurations.

Index Terms—Self-synchronization, Consensus, Localization, Positioning, ToA, Iterative Localization, Distributed Localization.

# I. INTRODUCTION

C ELLULAR NETWORKS have progressively increased and improved the functionality on the edge of the network such as the geo-localization service. For such a functionality, Positioning Reference Signals (PRS) are included in the protocols to support device localization based on the estimation of the signal Time-of-Arrival (ToA) [1]. The use of Ultra Dense Networks (UDN) and small cells increases the complexity of centralized localization approaches. This is due to the massive amount of data that needs to be exchanged between the cells and the central processing unit. Such architectures create bottlenecks and demand high communication bandwidths. In that context, fast and low-data-exchange distributed localization approaches are strongly desired.

Manuscript received May 10, 2021; revised October 22, 2021; accepted January 27, 2022.

E. Pocoma, F. Quitin, P. De Doncker, and F. Horlin are with the Université Libre de Bruxelles, Brussels 1050, Belgium (e-mail:epocomac@ulb.be; fquitin@ulb.ac.be; philippe.dedoncker@ulb.be; francois.horlin@ulb.be).

L. Vandendorpe is with the Université Catholique de Louvain, 1348 Ottignies-Louvain-la-Neuve, Belgium(e-mail:luc.vandendorpe@uclouvain.be) Several ToA-based localization approaches can be found in the literature. The Direct Position Estimation (DPE) algorithm has the best performance of all. It gathers all the received baseband signals in a processing node known as Fusion Centre (FC), and estimates the transmitter position using grid search<sup>1</sup> [2], [3]. A two-step algorithm was proposed to reduce the amount of communicated data [4]. It consists in a range estimation step at each Base Station (BS) followed by a multi-lateration step at the FC. However, it has been analytically demonstrated in [5] that a direct approach like the DPE algorithm outperforms any two step approach. Mainly because only local information is available at the BS for the range estimation step; whereas all the information is available in the DPE case.

1

The Iterative Position Estimation (IPE) algorithm proposed to iterate sequentially over the two steps with the goal of refining the range estimate. It was seen in [6] that the performance of the IPE algorithm gets very close to the one of the DPE algorithm. Therefore, the IPE algorithm also outperforms the two-step approach. Furthermore, the IPE algorithm has been extended for different case scenarios like: time-misaligned transmitters [7], rich multipath [8], angle-of-arrival based localization and others [9], [10]. Nonetheless, the IPE approach presents two main shortcomings. First, it is still a centralized approach. Second, the complexity at the FC grows considerably for a dense network. Therefore, in this paper we propose a fully distributed localization algorithm that overcomes these limitations.

Different approaches have been studied to achieve distributed localization. Specially, there are two widely adopted approaches: distributed optimization and distributed consensus. Different studies regarding the first approach can be found in [11], [12] and [13]. A distributed estimation framework is introduced in [11], which is based on Alternating Direction Method of Multipliers (ADMM). The resulting processing approach is not homogeneous among the nodes, since the information is gathered and fused/averaged in a few nodes denoted as bridges. In [12], [13], the distributed localization is done by explicitly solving the optimization problem by ADMM or Primal-Dual Method of Multipliers (PDMM). The second approach, i.e., distributed consensus, computes the average of the final estimated parameter in a fully distributed fashion [14], [15]. It has been seen that it decreases the localization error after the

<sup>1</sup>The DPE algorithm compares the baseband signals received by each Base Station to the signals that would theoretically be received if the transmitter was at a particular position.

0018-9545 (c) 2021 IEEE. Persona material har an orthoginar participation of the provided internation. Authorized licensed use limited to: Vrije Universiteit Brussel. Downloaded on February 08,2022 at 11:47:40 UTC from IEEE Xplore. Restrictions apply.

multi-lateration step [14]. In [15] the localization is done by means of an iterative process such as the Gauss-Newton algorithm, where the intermediate estimates are averaged in a distributed fashion at every single iteration.

In the already mentioned studies, the optimization and consensus approaches were used after the range estimation step. Therefore, the localization error of both approaches is lower bounded by the one of the two-step localization approach. Moreover, similar to the case of the IPE algorithm, it is clear that in order to outperform the two-step approach, the distributed solution should be implemented somehow before the second step. Unfortunately, the localization based on distributed optimization can be very challenging when it is referred to the baseband signals. In such a case, the final objective function is highly non-convex with several maximas and minimas in the range domain. Therefore, we propose here to make use of distributed consensus. However, we make use of a more general case such as the self-synchronization mechanism<sup>2</sup> introduced in [16].

In a nutshell, the self-synchronization mechanism associates a state variable to each node. Such state variables conform an autonomous dynamical system that converges to a common value for all state variables [16]. The convergence is guaranteed for very relaxed conditions. In fact, the only requirement for convergence is to have a strongly connected network. The convergence is guaranteed even for noisy and/or unreliable links, requiring a strongly connected network only on average over time [17], [18]. On the one hand, zero order consensus converges directly on the state variables. On the other hand, first order consensus converges on the first order derivative of the state variables. The main advantage of the first order over the zero order consensus is that it presents a better rejection to coupling noise [17]. The self-synchronization mechanism relies on the exchange of state variables among the nodes. Therefore, the coupling noise is introduced to the system when the communication of state variables involves approximation due to reconstruction or quantization [19].

Finally, the contributions in this paper are fourfold and can be summarized as follows:

- We propose a Self-Synchronization Position Estimation (SSPE) algorithm that works at the baseband signal level, i.e., we average the received signal log-likelihoods. Such an approach reaches consensus for the posterior distribution of the transmitter position rather than directly on the final position estimates. It also means that we are able to achieve the performance of the DPE algorithm in a distributed fashion.
- We show analytically that both IPE and the proposed SSPE algorithms exploit the same self-synchronization mechanism. Therefore, the improvements done for the IPE algorithm are directly applicable to our proposed approach thanks to that relationship. Besides, we choose a particular study case of localization of a time misaligned transmitter to show numerically that IPE and SSPE yield

very similar results. In short, the IPE approach can be considered as a particular case of the SSPE framework under the assumption of a fully connected network and a specific set of parameters. Furthermore, SSPE is a distributed algorithm, whereas IPE is centralized.

- The proposed algorithm as such, still demands high communication bandwidths, due to the grid-search nature of the algorithm. Therefore, we propose to compress the state variables with the radial, angular, and time misalignment first two order moments. Such a compression enables us to exchange only seven intermediate parameters among the BSs.
- Lastly, we assess and compare the performance of the developed algorithms by means of Monte Carlo simulations. We show and discuss the effects of the compression of state variables over performance of the proposed algorithm. We also present a convergence analysis for logical network configurations different from a fully connected one.

The paper is organized as follows. Section II first introduces the signal model. Section III secondly describes the centralized DPE algorithm and the proposed distributed SSPE algorithm. Afterwards, it describes the proposed compression and reconstruction for the state variable matrices and its effects. Section IV states the relationship between IPE and SSPE algorithms. Section V defines the performance metrics. Section VI assesses numerically and discusses the such performance metrics. Finally, Section VII concludes the paper.

#### **II. SYSTEM MODEL**

We assume a static transmitter that is simultaneously connected to N BSs in a cellular network, which operates with the Orthogonal Frequency-Division Multiplexing (OFDM) modulation. The OFDM modulation splits the communication bandwidth into orthogonal sub-carriers that are allocated to data or pilot symbols. A Cyclic Prefix (CP) is added to each block of transmitted symbols to maintain the orthogonality between the sub-carriers even in the presence of channel time dispersion. We consider that P equispaced pilot sub-carriers are allocated over the communication bandwidth. Moreover, the frequency difference between two consecutive pilot sub-carriers is constant and referred to as  $\Delta f$ .

To better explain the following steps of the proposed algorithm, we define two sets of parameters. We define  $\gamma_{n_i}$  and  $\gamma_c$  as the set of nuisance and common parameters respectively. The set of nuisance parameters  $\gamma_{n_i}$  is intrinsic to BS-*i*. Hence, such parameters are only meaningful for the particular BS-*i* and not for the rest of the network. The set of common parameters  $\gamma_c$  is related/common to all the BSs. Therefore, they are meant to be estimated using the whole information of the network.

We consider the same case scenario studied in [7] where the IPE was improved for localization of a time-misaligned transmitter. Notice that it is considered that all BSs are perfectly time synchronized. Therefore, we take into account only the time-offset  $t_0$  between the transmitter and the network. Furthermore, we choose such a scenario to show

<sup>&</sup>lt;sup>2</sup>Self-synchronization mechanism will converge to the same value as the distributed average consensus when any two neighboring nodes can receive/send data from/to each other.

IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. XX, NO. X, FEBRUARY 2022

that the improvements done for the centralized IPE algorithm are easily applicable to our distributed-proposed approach. Consequently and similar to [7], we focus our study to jointly estimates the transmitter position (x, y) and the time offset  $t_0$ .

We assume a propagation channel that has a dominant Line-of-Sight (LOS). Therefore, the channel is characterized by a time delay  $\tau_i$  between the transmitter position and BS-*i*. The propagation delay is related to the distance  $d_i$  between transmitter and BS-*i* by  $\tau_i = d_i/c$  where *c* is the propagation velocity. Therefore, the time of arrival  $t_i$  observed by BS-*i* is the addition of the time delay and time offset, i.e.,

$$t_i(x, y, t_0) = \tau_i(x, y) + t_0, \tag{1}$$

where  $t_i$  is assumed to be shorter than the CP duration (which is a reasonable assumption for typical system parameters). For convenience and without loss of generality, we rewrite (1) in terms of pseudoranges by simply multiplying both sides of the equation by the propagation velocity, which yields

$$\delta_i(x, y, \delta_0) = d_i(x, y) + \delta_0. \tag{2}$$

We refer to  $\delta_0$  as range offset, which is directly linked to the time offset  $t_0$  through  $\delta_0 = c t_0$ . Consequently, the set of common parameters considered in this work is

$$\gamma_c = [x, y, \delta_0]. \tag{3}$$

Regarding the nuisance parameters, the signal is affected by Carrier Frequency Offset (CFO), which is roughly estimated and compensated before any processing takes place. Nonetheless, there is a remaining CFO that can be approximated as a common phase offset  $\phi_i$ , affecting all sub-carriers in a given multi-carrier block (OFDM symbol) [7]. Besides, we model the amplitude of the received signal with a coefficient  $a_i$  that is rician distributed. Hence, the set of nuisance parameters at BS-*i* is considered to be

$$\gamma_{n_i} = [a_i, \phi_i]. \tag{4}$$

Consequently, the received signal at BS-i over pilot sub-carrier-p can be modeled as

$$r_{i,p} = u_i(\gamma_{n_i})s_{i,p}(\gamma_c) + w_{i,p},\tag{5}$$

where  $r_{i,p}$  and  $w_{i,p}$  are the received signal and corrupting noise respectively, for pilot sub-carrier-p at BS-i. The noise is assumed to be independent zero mean circularly symmetric complex Gaussian of variance  $\sigma_{w_i}^2$  known at the BS. In addition,  $u_i$  and  $s_{i,p}$  are defined as

$$u_i(\gamma_{n_i}) = a_i e^{\phi_i},\tag{6}$$

$$\beta_{i,p}(\gamma_c) = q_p e^{-jp\zeta\delta_i(x,y,\delta_0)},\tag{7}$$

where  $q_p$  is the symbol transmitted at pilot sub-carrier-*p*. In addition, the constant  $\zeta$  is used to simplify the notation and it is defined as  $\zeta = \frac{2\pi\Delta f}{c}$ . Notice that the product of  $u_i$  and  $s_{i,p}$  represents the free-noise part of the model. Moreover,  $u_i$  and  $s_{i,p}$  model respectively the contribution of the nuisance and common parameters at BS-*i* and pilot sub-carrier-*p*. To simplify the notation in the rest of the paper, we write only  $\delta_i$ 

instead of  $\delta_i(x, y, \delta_0)$ ; however, we mention such dependence whenever it is needed.

Finally and similarly to the system model adopted in [7] and [6], a vector model is constructed at each BS-iby stacking all the received pilot symbols as

$$\mathbf{r}_i = u_i(\gamma_{n_i})\mathbf{s}_i(\gamma_c) + \mathbf{w}_i,\tag{8}$$

3

with

$$\mathbf{r}_{i} = [r_{i,1}, ..., r_{i,P}]^{T} ; \quad \mathbf{w}_{i} = [w_{i,1}, ..., w_{i,P}]^{T}, \quad (9)$$
$$\mathbf{s}_{i}(\gamma_{c}) = [q_{1}e^{-j1\zeta\delta_{i}}, ..., q_{P}e^{-jP\zeta\delta_{i}}]^{T}. \quad (10)$$

#### **III. SELF-SYNCHRONIZATION POSITIONING ESTIMATION**

To better explain the steps of the proposed SSPE algorithm, we start by describing a centralized DPE algorithm that jointly estimates the transmitter position and the range offset. Besides, we show how to carry out the nuisance parameters to obtain a range likelihood dependent only on the set of common parameters. Afterwards, we describe how the SSPE algorithm approximates such centralized approach in a distributed fashion. Namely, we describe the use of the self-synchronization mechanism, as well as, how to compress the state variables.

# A. Centralized Direct Position Estimation

The DPE algorithm estimates the transmitter position based on all the received signals. Hence, each BS-*i* should communicate its received baseband signal  $\mathbf{r}_i$  to a central node denoted as FC. The posterior Probabilistic Density Function (PDF) of the common parameters  $\gamma_c$  given by the received signals is

$$p(\gamma_c | \mathbf{r}_1, \dots \mathbf{r}_N) = C_{\gamma_c} \prod_{i=1}^N p_i(\mathbf{r}_i | \gamma_c) p(\gamma_c), \qquad (11)$$

where  $C_{\gamma_c}$  is a normalization factor ensuring that the integral of the posterior PDF in the considered search grid is 1.  $p_i(\mathbf{r}_i|\gamma_c)$  is the range likelihood of the received signal and  $p(\gamma_c)$  is the prior PDF of the common set of parameters. The considered search grid is denoted as S. Moreover, Sis defined as a multi-dimensional interval defined by the Cartesian product of  $x \in [x_{min}, x_{max}], y \in [y_{min}, y_{max}]$ , and  $\delta_0 \in [\delta_{0_{min}}, \delta_{0_{max}}]$ .

We consider that common parameters are mutually independent and uniformly distributed on S. Consequently,  $p(\gamma_c)$  is a uniform PDF given by

$$p(\gamma_c) = \begin{cases} \frac{1}{\Delta_x \Delta_y \Delta_{\delta_0}} & \gamma_c \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}.$$
 (12)

where  $\Delta_x$ ,  $\Delta_y$  and  $\Delta_{\delta_0}$  are similarly defined as  $\Delta_{\psi} = \psi_{max} - \psi_{min}$ .

In addition, we can define the likelihood of the received signal at BS-i based on (8). More precisely, such likelihood can be modeled as a Gaussian distribution as

$$p_i(\mathbf{r}_i|\gamma_{n_i},\gamma_c) = C_{\mathbf{r}_i} e^{-\frac{1}{\sigma_{w_i}^2}[\mathbf{r}_i - u_i \mathbf{s}_i]^H[\mathbf{r}_i - u_i \mathbf{s}_i]}, \quad (13)$$

<sup>0018-9545 (</sup>c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications\_standards/publications/rights/index.html for more information. Authorized licensed use limited to: Vrije Universiteit Brussel. Downloaded on February 08,2022 at 11:47:40 UTC from IEEE Xplore. Restrictions apply.

IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. XX, NO. X, FEBRUARY 2022

where  $(.)^{H}$  represents the Hermitian transpose and  $C_{\mathbf{r}_{i}}$  is a normalization factor. Notice that the dependence of  $u_{i}$  and  $s_{i}$  on  $\gamma_{n_{i}}$  and  $\gamma_{c}$  respectively, is not written to simplify the notation. Clearly the likelihood of the received signal shown in (13) depends on both the nuisance parameters  $\gamma_{n_{i}}$  and common parameters  $\gamma_{c}$ . Therefore the PDF  $p_{i}(\mathbf{r}_{i}|\gamma_{c})$  is obtained by marginalizing the nuisance parameters  $\gamma_{n_{i}}$  from (13) as:

$$p_i(\mathbf{r}_i|\gamma_c) = \int_{-\infty}^{\infty} p_i(\mathbf{r}_i|\gamma_{n_i}, \gamma_c) p(\gamma_{n_i}) d\gamma_{n_i}$$
(14)

where  $p(\gamma_{n_i})$  is the distribution of the nuisance parameters. Namely,  $p(\gamma_{n_i})=p(\phi_i)p(a_i)$  where  $\phi_i$  is assumed to be uniformly distributed for  $\phi_i \in [-\pi, \pi]$ , and  $a_i$  is assumed to be rician distributed. Notice that (14) is a compressed notation for a multiple integral (one integral per nuisance parameter). The extended version of (14) can be found in (50) in Appendix A.

Finally, with the assumption of  $\phi_i$  being uniformly distributed, the expression shown in (14) can be further simplified to

$$p_i(\mathbf{r}_i|\gamma_c) = C_{\phi_i} \int_{-\infty}^{\infty} J_0(z) e^{\frac{-a_i^2}{\sigma_{w_i}^2} \mathbf{s}_i^H \mathbf{s}_i} p(a_i) da_i, \qquad (15)$$

where  $z=-j\frac{2a_i}{\sigma_{w_i}^2}|\mathbf{r}_i^H\mathbf{s}_i|$  and  $C_{\phi_i}$  is a constant term. Notice that for the sake of clarity, the dependence of  $\mathbf{s}_i$  on the common parameters  $\gamma_c$  is not written. Moreover, the variable z depends on  $\gamma_c$  and the amplitude coefficient  $a_i$ , i.e.,  $z=z(\gamma_c, a_i)$ . The variable z does not depend on  $\phi_i$  due to the fact that the integral with respect to  $\phi_i$  has been replaced by a Bessel function of order zero  $J_0(z)$  (See Appendix A). Such use of the Bessel function means a reduction of complexity, and thus a faster numerical integration when computing  $p_i(\mathbf{r}_i|\gamma_c)$ .

#### B. SSPE Algorithm

The goal of our proposed SSPE algorithm is to approximate, in a distributed way, the posterior PDF of  $\gamma_c$  defined in (11). The self-synchronization mechanism computes the average of the measurements taken by each node in the network. Therefore, we express the posterior PDF of  $\gamma_c$  in terms of averages as follows.

First, we take the logarithm of (11), which yields the log-posterior distribution written as

$$\log\left(p(\gamma_c|\mathbf{r}_1,...\mathbf{r}_N)\right) = \sum_{i=1}^N \mathcal{L}_i(\mathbf{r}_i|\gamma_c) + b, \qquad (16)$$

where the constant  $b = \log(p(\gamma_c)) + \log(C_{\gamma_c})$  considers the normalization factor and the uniformly distributed prior.  $\mathcal{L}_i(\mathbf{r}_i|\gamma_c)$  is the log-likelihood of the received signal  $\mathbf{r}_i$  at BS-*i*. Furthermore,  $\mathcal{L}_i(\mathbf{r}_i|\gamma_c)$  can be computed by taking the logarithm of (14) or (15), i.e., after the marginalization of the nuisance parameters  $\gamma_{n_i}$ .

Second, we simply multiply and divide the sum of range log-likelihoods by the number of considered BSs as follows

$$\log\left(p(\gamma_c|\mathbf{r}_1,...\mathbf{r}_N)\right) = N\left(\frac{1}{N}\sum_{i=1}^N \mathcal{L}_i(\mathbf{r}_i|\gamma_c)\right) + b.$$
(17)

It is easy to see that the first term is nothing else than the average of the received signal log-likelihoods. Consequently, we make two important remarks regarding (17). First, the constant term b can be omitted if the posterior distribution is normalized again after taking the exponential to compute the corresponding PDF. Second, the average of the received signal log-likelihoods can be computed in a fully distributed fashion with the help of the self-synchronization mechanism introduced in [16]. It is worth noting that (17) holds for every single point in the scene S. Hence, the self-synchronization mechanism is applied to each single point individually. Each BS-i will have a state variable matrix  $\mathbf{v}_i$  that will evolve at iteration-k accordingly to

4

$$\Delta \mathbf{v}_i[k] = \mathcal{L}_i(\mathbf{r}_i|\gamma_c) + \beta \sum_{j=1}^N \alpha_{i,j}(\mathbf{v}_j[k] - \mathbf{v}_i[k]), \quad (18)$$

where:

- $\alpha_{i,j}$  is 1 if node-*i* and node-*j* are connected and 0 otherwise.
- $\mathbf{v}_j[k]$  is the state variable matrix of node-*j* communicated to node-*i* at iteration-*k*. Notice that at iteration *k*=0 the initial value is  $\mathbf{v}_i[0]=\mathbf{0}$  for all nodes.
- $\beta$  is a constant defined as the control loop gain.

It has been well studied in [16], [18], [17], and [20] that the dynamical system shown in (18) will asymptotically converge for the increments of state variables to  $\Delta \mathbf{v}_i^*$ . To define the convergence value we first need to define three concepts: strong connectivity, undirected network and balanced network. We provide a condensed and intuitive definition of these terms in this paper. However, we refer the reader to [16]-*Directed Graphs* for a formal and detailed definition.

Node-*i* is reachable from node-*j* if there is a sequence of adjacent nodes, i.e. a path, which starts with node-*i* and ends with node-*j*. With that remark we proceed to define the following concepts. First, a network is strongly connected if every node-*i* is reachable from every other node-*j*. Second, a network is undirected if any node can receive and send messages from and to all its adjacent nodes. Third, a network is balanced if any node sends and receives the same number of messages. Lastly, notice that if a network is undirected it is also balanced.

The convergence is ensured if and only if the network is strongly connected. In addition, we restrict our study to balanced networks to ensure that the convergence value is, as desired, the average of the received signal log-likelihoods. Consequently, considering that the network is strongly connected and balanced the dynamical system converges to

$$\Delta \mathbf{v}_i^* = \Delta \mathbf{v}_i[k \to \infty] = \frac{1}{N} \sum_{j=1}^N \mathcal{L}_j(\mathbf{r}_j | \gamma_c).$$
(19)

As a result, based on (17) and (19), we can approximate the posterior distribution at each BS-i for iteration-k as

$$\hat{p}_i(\gamma_c | \mathbf{r}_1, ..., \mathbf{r}_N)[k] = C_{\Delta \mathbf{v}_i} e^{N \Delta \mathbf{v}_i[k]}, \qquad (20)$$

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications\_standards/publications/rights/index.html for more information. Authorized licensed use limited to: Vrije Universiteit Brussel. Downloaded on February 08,2022 at 11:47:40 UTC from IEEE Xplore. Restrictions apply.

IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. XX, NO. X, FEBRUARY 2022

Algorithm 1 SSPE Algorithm	Se
1: Each BS- <i>i</i> constructs $\mathcal{L}_i(\mathbf{r}_i \gamma_c)$ for each point in the search	the u
grid based on the received signal $\mathbf{r}_i$ .	
2: Each BS- <i>i</i> initializes $\mathbf{v}_i = 0$	
3: for iteration $k=1,2,\ldots$ do	
4: <b>for</b> BS- $i=1, 2, \ldots$ in parallel <b>do</b>	wher
5: Compute $\Delta \mathbf{v}_i[k]$ using (18)	inter
6: Estimate transmitter position using (21)	$(\delta_i, \ell$
7: Update $\mathbf{v}_i[k+1] = \mathbf{v}_i[k] + \Delta \mathbf{v}_i[k]$	are 1
8: Communicate $\mathbf{v}_i[k+1]$ to neighboring BSs	obse
9: Receive $\mathbf{v}_i[k+1]$ from neighboring BSs	are c
10: end for	
11: end for	

where the constant  $C_{\Delta \mathbf{v}_i}$  is a normalization factor ensuring that the integral of (20) in the scene S is 1. Notice that  $N\Delta \mathbf{v}_i$ will asymptotically converge to the sum of the received signal log-likelihoods as shown in (19). Hence, the expression in (20) is nothing else than an approximation that asymptotically converges to the posterior distribution defined in (11). Based on (20), it is possible to numerically compute an estimate of the transmitter position  $(\hat{x}, \hat{y})$  following the Minimum Mean Squared Error (MMSE) estimator (as it was done in [6], [7]). Notice that each BS-*i* is able to compute such estimation as:

$$\hat{\gamma}_{c_i}[k] = \mathbb{E}[\gamma_c | \mathbf{r}_1, ..., \mathbf{r}_N] = \int_{\mathcal{S}} \gamma_c \, \hat{p}_i(\gamma_c | \mathbf{r}_1, ..., \mathbf{r}_N)[k] d\gamma_c.$$
(21)

Lastly, the SSPE algorithm can be summarized in the pseudo-code Algorithm 1. It describes how the dynamical system defined in (18) evolves. In addition, it is worth noticing that the next value of state variables  $\mathbf{v}_i[k+1]$  is computed based on the increment  $\Delta \mathbf{v}_i[k]$  and its current value  $\mathbf{v}_i[k]$  as shown in *step-7*.

#### C. Compression and Reconstruction of State Variables

The self-synchronization mechanism requires to exchange state variables between neighboring nodes. In Section-III-B, we propose to apply the self-synchronization mechanism to every single point in the scene. This means that  $\mathbf{v}_i$  is a matrix of same dimensions as S. Therefore, the exchange of such state variables  $\mathbf{v}_i$  demands high communication bandwidth, making the proposed algorithm impractical. For such a reason, we propose to compress the state variables  $\mathbf{v}_i$ . We compute the first two order moments in pseudo-range  $\delta_i$ , angle  $\theta_i$  and range-offset  $\delta_{0_i}$  following the next two steps.

First, the state variables  $\mathbf{v}_i$  operate with the range log-likelihood of BS-*i* as seen in (18). Consequently, we propose to define a probability distribution  $p_{\mathbf{v}_i}$  associated to each state variable  $\mathbf{v}_i$ . Such a probability distribution is defined at iteration k as

$$p_{\mathbf{v}_i}[k] = C_{\mathbf{v}_i} e^{\mathbf{v}_i[k]},\tag{22}$$

where  $C_{\mathbf{v}_i}$  is a normalization factor ensuring that the integral of  $p_{\mathbf{v}_i}$  in the scene S is 1.

Second, we compute the moments of  $p_v$  using the law of the unconscious statistician (LOTUS) rule as

$$\mathbb{E}_{p_{\mathbf{v}_i}}\left[f(\gamma_c)\right] = \int_{\mathcal{S}} f(\gamma_c) \, p_{\mathbf{v}_i}[k] \, d\gamma_c \tag{23}$$

5

where  $f(\gamma_c)$  is a function defined in the multi-dimensional interval S. We compute the first two order moments for  $(\delta_i, \theta_i, \delta_0)$ . Notice that  $\delta_0$  is the range-offset,  $\delta_i$  and  $\theta_i$ are respectively the pseudo-range and angle at which BS-*i* observes the transmitter. In addition, the parameters  $(\delta_i, \theta_i, \delta_0)$ are defined as

$$\delta_i(x, y, \delta_0) = d_i(x, y) + \delta_0, \qquad (24)$$

$$\theta(x,y) = \arctan\left(\frac{y-y_i}{x-x_i}\right).$$
(25)

On the one hand, each BS has pseudo-range  $\delta_i$  information that comes from the received baseband signal. On the other hand, the angle  $\theta_i$  is extracted from the shared information between BSs. Therefore, when the transmitter is surrounded by the BSs, as we consider in our study case, the cross-correlation between pseudo-range  $\delta_i$  and angle  $\theta_i$  is very small. Hence, we choose to assume that the pseudo-range  $\delta_i$  and angle  $\theta_i$ are independent.

The pseudo-range  $\delta_i$  and range offset  $\delta_0$  are clearly correlated as shown in (24). Consequently, the first order moments can be computed as

$$\hat{\delta}_{i}[k] = \mathbb{E}_{p_{\mathbf{v}_{i}}}\left[\delta_{i}\right]; \, \hat{\delta}_{0}[k] = \mathbb{E}_{p_{\mathbf{v}_{i}}}\left[\delta_{0}\right]; \, \hat{\theta}_{i}[k] = \mathbb{E}_{p_{\mathbf{v}_{i}}}\left[\theta\right], \quad (26)$$

and the second order moments as

$$\Sigma_{\hat{\delta}_{i,0}}[k] = \mathbb{E}_{p_{\mathbf{v}_i}}\left[\Delta_{\delta_{i,0}}\Delta_{\delta_{i,0}}^T\right]$$
(27)

$$\sigma_{\hat{\theta}_i}^2[k] = \mathbb{E}_{p_{\mathbf{v}_i}}\left[\left(\theta - \hat{\theta}_i[k]\right)^2\right]$$
(28)

where  $\Delta_{\delta_{i,0}} = [\delta_i - \hat{\delta}_i[k], \delta_0 - \hat{\delta}_0[k]]^T$  and  $\Sigma_{\hat{\delta}_{i,0}}$  is a 2x2 matrix. Finally, the exchange of parameters happens at *step-8* and *step-9* of Algorithm 1. Only seven parameters  $(\hat{\delta}_i, \hat{\delta}_0, \Sigma_{\hat{\delta}_{i,0}}, \hat{\theta}_i, \sigma_{\hat{\theta}_i}^2)$  are exchanged between neighboring nodes instead of all the elements of the state variable matrix  $\mathbf{v}_i$ . In *step-9* of Algorithm 1, BS-*i* receives the

$$\hat{\mathbf{v}}_{j}[k] = -\frac{1}{2} \Delta_{\delta_{j,0}}^{T} \Sigma_{\hat{\delta}_{j,0}}^{-1} \Delta_{\delta_{j,0}} - \frac{1}{2\sigma_{\hat{\theta}_{j}}^{2}} (\theta - \hat{\theta}_{j})^{2}.$$
 (29)

#### D. Convergence Rate Analysis

parameters and reconstructs  $\mathbf{v}_i$  as

The associated dynamical system shown in (18) evolves at a certain convergence rate that is dependent on the communication parameters and the logical network configuration [21]. Similarly to the analysis done in [22], it could be shown that the state variables of the whole network will evolve as

$$\Delta \mathbf{V}[k] = \mathbf{\Phi}^{k-1} \mathbf{g},\tag{30}$$

with

$$\Delta \mathbf{V} = [\Delta \mathbf{v}_1, ..., \Delta \mathbf{v}_N]^T \quad ; \quad \mathbf{g} = [\mathcal{L}_1, ..., \mathcal{L}_N]^T, \quad (31)$$

where  $\Phi = I - \beta L$ , with I and L being the identity and Laplacian matrix respectively.

6

IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. XX, NO. X, FEBRUARY 2022

On the one hand, the matrix  $\Phi$  describes the relationship at iteration-k between the increment of state variables of the whole system  $\Delta V$  and the initial range log-likelihoods g. On the other hand, under the assumption of a balanced and strongly connected network, the convergence value for the whole dynamical system is the average of range log-likelihoods, i.e.,

$$\Delta \mathbf{V}^* = \frac{1}{N} \mathbf{1} \mathbf{1}^T \mathbf{g},\tag{32}$$

where 1 is a N-vector of ones. Consequently, we can find an upper and lower bound on the evolution of  $\Delta V$  by subtracting (32) from (30) and taking the norm as follows

$$\left|\left|\Delta \mathbf{V}[k] - \Delta \mathbf{V}^*\right|\right| \le \lambda_{max}(\mathbf{W}_k) \left|\left|\mathbf{g}\right|\right|,\tag{33}$$

where  $\lambda_{max}(\mathbf{W}_k)$  is the biggest eigenvalue of  $\mathbf{W}_k$ , which is defined as  $\mathbf{W}_k = \Phi^{k-1} - \frac{1}{N} \mathbf{1} \mathbf{1}^T$ . In short, based on (33), it is clear that the smaller  $\lambda_{max}(\mathbf{W}_k)$  is, the closer the network is to the convergence value. Besides, it was shown in [20] that  $\lambda_{max}(\mathbf{W}_k)$  is associated to the network configuration as

$$\lambda_{max}(\mathbf{W}_k) = (1 - \beta \lambda_2(\mathbf{L}))^{k-1}, \qquad (34)$$

where  $\lambda_2(\mathbf{L})$  is the second smallest eigenvalue of the graph Laplacian, also known as Fiedler value or algebraic connectivity [23].

On the one hand, notice that the convergence bound described in (33) is given for the increment of state variables  $\Delta \mathbf{v}_i$ . Which in our case represent the log-posterior PDF of the transmitter position. On the other hand, notice that we are interested on the convergence rate of the final expected values of  $\gamma_c$ . Which have direct relationship to the approximated log-posterior as show in (21). Therefore, we can still make use of such convergence-rate analysis to address the convergence of the SSPE algorithm, as we will see in Section VI-D.

# IV. SSPE AND IPE ALGORITHM RELATIONSHIP

In this section we show the relationship between IPE and the proposed SSPE algorithm. In a nutshell, we prove that both algorithms exploit the self-synchronization mechanism. Hence, the benefits and improvements done for the IPE algorithm can be directly applied to the SSPE algorithm.

To better explain such relationship, we briefly introduce the IPE algorithm. However, a more detailed derivation can be found in Appendix B. In brief, the IPE can be summarized as the iteration of the two following steps:

• Intermediate Parameter Estimation: At BS-*i*, it estimates the intermediate parameters  $\psi_i$  following a Bayes framework. The log-posterior distribution of  $\psi_i$  is shown in (35) (Notice that the expression is in log-domain), where  $\mathcal{L}_i(\mathbf{r}_i|\psi_i)$  is the range log-likelihood of the received signal.  $\mathcal{P}_i(\psi_i)$  is a log-prior that is updated at each iteration by the FC, and k is the iteration index.

$$\mathcal{P}_i(\psi_i|\mathbf{r}_i)[k+1] = \mathcal{L}_i(\mathbf{r}_i|\psi_i) + \mathcal{P}_i(\psi_i)[k]$$
(35)

• *Prior Refinement*: At the FC, the prior  $\mathcal{P}_i(\psi_i)$  is updated based on the information sent by all the other BSs (See Appendix B).

To show that the IPE makes use of the self-synchronization mechanism, we try to express (35) in a similar way to the dynamical system equation shown in (18). To do so, we perform two operations on (35). First, we subtract  $\mathcal{P}_i(\psi_i|\mathbf{r}_i)[k]$  from both sides of the expression, i.e., we subtract the log-posterior at iteration k. Second and without changing the IPE framework, we consider the case where  $\psi_i = \gamma_c$ , i.e., we consider that the intermediate parameters are the final estimated parameters. Under such assumption we can change the dependency of (35) to obtain:

$$\Delta \mathcal{P}_i(\gamma_c | \mathbf{r}_i)[k] = \mathcal{L}_i(\mathbf{r}_i | \gamma_c) + \mathcal{P}_i(\gamma_c)[k] - \mathcal{P}_i(\gamma_c | \mathbf{r}_i)[k], \quad (36)$$

where  $\Delta \mathcal{P}[k]$  is defined as the forward increment, i.e.,  $\Delta \mathcal{P}[k] = \mathcal{P}[k+1] - \mathcal{P}[k]$ . Essentially, expression (36) yields the evolution of the posterior probability of  $\gamma_c$  throughout the iterations. A more detailed derivation of (36) can be found in Appendix B.

Expression (36) is very similar to the self-synchronization expression used for localization defined in (18). Furthermore, by distributing  $\alpha_{i,j}$  and the summation in the last term, (18) can be written as

$$\Delta \mathbf{v}_i[k] = \mathcal{L}_i(\mathbf{r}_i|\gamma_c) + \beta \sum_{j=1}^N \alpha_{i,j} \mathbf{v}_j[k] - \mathbf{v}_i[k] \beta \sum_{j=1}^N \alpha_{i,j}.$$
 (37)

By comparing (36) and (37) we can make the following remarks term by term:

- It can be seen easily in the left hand side terms that the state variable  $\mathbf{v}_i$  is equivalent to the log-posterior  $\mathcal{P}_i(\gamma_c | \mathbf{r}_i)$ .
- The log-likelihood term of the received signal is the same for both expressions and it remains constant through out iterations.
- If we choose arbitrarily the value of the control loop gain as  $\beta^{-1} = \sum_{j=1}^{N} \alpha_{i,j}$ , the last term in (37) is equivalent to the last term in (36).
- The second term of the right-hand side of (37) combines the information of all BSs connected to BS-*i*. Similarly, the log-prior term  $\mathcal{P}_i(\gamma_c)$  in (36) is also computed based on the information of all the other BSs, as can be seen in (55) in Appendix B.

Consequently, the IPE can be seen as a particular case of the proposed SSPE framework since both of them exploit the self-synchronization mechanism. In summary, the conditions for such equivalency are twofold: a particular value of control loop gain  $\beta$  and a fully connected network. As a result, the SSPE extends the IPE localization principle from a centralized localization to a fully distributed scenario.

To the best of our knowledge there is no IPE convergence analysis proposed in the literature. Conversely, in the SSPE approach, we can rely on the results from the self-synchronization literature to study the convergence of the SSPE approach for different network configurations. Due to such relationship between IPE and SSPE algorithms, most of the IPE studies and its improvements can be extended to the proposed SSPE framework. For instance IPE was extended in [7] to consider a time offset between transmitter and BSs, which can be directly related to this paper. In [8], IPE was

IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. XX, NO. X, FEBRUARY 2022

extended for rich multipath scenarios which can be addressed in the SSPE framework by modifying the likelihood of the received signal.

# V. PERFORMANCE ANALYSIS

The proposed SSPE algorithm is fully distributed, whereas the DPE and IPE approaches are centralized. Such a characteristic is the most distinctive among them. However, it is also important to quantify the performance of the proposed approach and compare it with other state-of-the-art approaches. In this section, we define three performance categories: Localization, Computational Complexity and Communication Overhead. Furthermore, we assess the performance of the DPE, IPE and the proposed SSPE algorithm.

#### A. Localization Performance

The localization performance is assessed in terms of Mean Distance Error (MDE) and Mean Range Offset Error (MROE). The MDE and MROE are indicatives of possitioning and synchronization error, respectively. Both metrics are defined for a particular Signal-to-Noise-Ratio (SNR) value as

$$MDE = \frac{1}{N_s} \sum_{n=1}^{N_s} \sqrt{(\hat{x}_n - x_n)^2 + (\hat{y}_n - y_n)^2}, \qquad (38)$$

$$MROE = \frac{1}{N_s} \sum_{n=1}^{N_{sim}} |\hat{\delta}_{0,n} - \delta_{0,n}|, \qquad (39)$$

where  $N_s$  is the total number of realizations. Besides, the coordinates  $(\hat{x}_n, \hat{y}_n)$  and  $(x_n, y_n)$  are the estimated and true transmitter positions respectively for realization n. Similarly,  $\hat{\delta}_{0,n}$  and  $\delta_{0,n}$  are the estimated and true range offset for realization n.

#### B. Computational Complexity

The complexity analysis for the DPE and IPE algorithms was addressed in [7]-Section-IV. Therefore, in this paper we only state the changes to those expressions and we refer the reader to the original paper for a detailed derivation. The complexity analysis in [7] is given as the number of real multiplications in the whole network. Nevertheless, we decide to separate and clearly state the complexity at each individual BS and FC.

The algorithms to be compared are grid-search based. Therefore, we define the number of elements in the final grid as  $N_{xy\delta_0} = N_x N_y N_{\delta_0}$ , where  $N_x$ ,  $N_y$  and  $N_{\delta_0}$  are the number of considered elements in the x,y and  $\delta_0$  axes. Likewise, the number of elements considered in the numerical marginalization of the signal amplitude  $a_i$  is denoted as  $N_a$ .

1) DPE: An approximation of the DPE complexity can be found in [7]-Eq.(33). Such an expression was derived taken into account the marginalization over the parameter  $\phi_i$ . However, such a marginalization is not needed anymore using (15). Consequently, the approximation of the DPE complexity considered in this paper is

$$C_{DPE}^{FC} \approx N(3P + N_a) N_{xy\delta_0}.$$
(40)

2) *IPE:* The IPE complexity was derived in [7]-Eq.(35). However, we separate the BS complexity from the FC complexity as

$$C_{IPE}^{BS} \approx (3P + N_a)N_\delta + N_\delta N_{it} \tag{41}$$

7

$$C_{IPE}^{FC} \approx N(N-1)N_{xy\delta_0}N_{it},\tag{42}$$

where  $N_{it}$  is the number of iterations and  $N_{\delta}$  is the number of elements considered in the pseudo-range grid.

3) SSPE: We analyze the pseudo-code given in Algorithm-1. At step-1, each BS computes the received signal log-likelihood. Similar to the DPE case, each BS needs 3P real multiplications to compute  $\mathbf{r}_i^H \mathbf{s}_i$ . Besides,  $N_a$  real multiplications are required to marginalize out the nuisance parameter  $a_i$  at every point in the final grid. Therefore, in total  $(3P+N_a)N_{xy\delta_0}$  real multiplications are needed at *step-1*. Notice that in [7] the complexity of expected values using the grid as well as addition and exponential operations are neglected. We also neglect those operations in this paper to do a fair comparison. Consequently, the complexities in steps 5, 6, 7 of Algorithm-1 are negligible compared to the final complexity. Step-7 also involves the exponential and expectation operations to perform the compression of state variables. Hence, their complexity is also neglected for the same reason. At every iteration, step-9 requires a grid reconstruction using (29) per each directly-connected neighbour. The first term in (29) requires three products per element in the  $N_{xy\delta_0}$  grid. Conversely, the second term requires only one product per element in the  $N_x N_y$  grid, since it does not depend on  $\delta_0$ . Lastly, an approximation of the SSPE complexity at each BS-i is given by

$$C_{i_{SSPE}}^{BS} \approx (3P + N_a) N_{xy\delta_0} + (3N_{xy\delta_0} + N_{xy}) |\mathcal{N}_i^{(1)}| N_{it},$$
(43)

where  $|\mathcal{N}_i^{(1)}|$  is the number of neighboring/directly-connected BSs to the BS-*i*.

## C. Communication Overhead

An analysis regarding the communication overhead was also presented in [7]-Section-IV. Nevertheless, the analysis is done in terms of the total number of exchanged bits. In distributed approaches, the processing relies on the exchange of parameters among the BSs; whereas in centralized approaches, the communication is done once or a few times but only between BSs and the FC. Therefore, we decided to analyze the number of received parameters per BS to achieve a fair comparison between distributed and centralized approaches.

1) DPE: Each BS transmits its received signal to the FC. Such a received signal is composed of P complex-value pilots. Each pilot is composed of a real and imaginary part. Therefore, the number of received parameters at the FC is

$$O_{DPE}^{FC} = 2PN. \tag{44}$$

2) *IPE:* At each iteration, the FC receives two parameters (pseudo-range mean and variance) from each BS. However, each BS receives two parameters (prior mean and variance) from the FC starting from the second iteration onward.

Therefore, the number of received parameters at the BS and FC respectively are

$$O_{IPE}^{BS} = 2(N_{it} - 1)$$
;  $O_{IPE}^{FC} = 2N_{it}N.$  (45)

Notice that the FC is the most congested node. Furthermore, the more BSs are present in the network, the more congested the FC is.

*3) SSPE:* As stated at the end of Section III-C, each BS receives seven parameters from every directly-connected BS. Therefore, the number of received parameters at BS-*i* is

$$O_{i_{SSPE}}^{BS} = 7|\mathcal{N}_{i}^{(1)}|N_{it} \tag{46}$$

#### **VI. SIMULATION RESULTS**

We consider a rectangular scene delimited by  $x \in [0m, 100m]$  and  $y \in [0m, 100m]$ , where N=8 BSs are located at the positions given in Table I (A graphical representation of the locations of the BSs is given in Fig. 4). The transmitter lies at arbitrary positions inside the rectangular scene and communicates with the BSs over a bandwidth of 40 MHz. At each BS, the processing is done using a single OFDM symbol containing P=64 equispaced pilots with  $\Delta f$ =312.5kHz. All BSs are perfectly time synchronized whereas the transmitter is not time synchronized to the BSs. Hence the time offset  $t_0$  is considered to be uniformly distributed in the interval  $t_0 \in [0ns, 250ns]$ , which in range offset means  $\delta_0 \in [0m, 75m]$  [1].

In addition, we consider a single path propagation channel, i.e., a strong LOS scenario. Therefore, according to the specification of ETSI TDL-E channel model [1], we draw the values of  $a_i$  from a Rician distribution of rice factor of 22dB. Such a rician factor is assumed to be known at the receiver. The common phase  $\phi_i$  is drawn from a uniform distribution in the range  $[-\pi, \pi]$ . The numerical integral to marginalize the amplitude  $a_i$  is assessed using  $N_a=10$  equispaced points. Such a grid is symmetrically placed around its mean considering a range of values of six times the standard deviation.

Similarly to [7] and [6], the SNR is defined as  $SNR=\sum_p |s_p|^2/(P\sigma_w^2)$ . In a real case scenario the BSs close to the transmitter will have higher SNR compared to the BSs at a further distance. However, we assume the SNR to be equal and known at all BSs. Such an assumption is done just for convenience to simplify the simulations.

To better show our results, we consider a fully connected network in Sections VI-A, VI-B, VI-C and VI-F since we want to compare the SSPE to the IPE algorithm. Conversely we consider network configurations different for a fully connected one for Section VI-D. Lastly, the MDE and MROE are averaged over  $N_s$ =200 transmitter positions, channel and noise realizations for each single SNR value.

TABLE I BSs locations

BS-idx	1	2	3	4	5	6	7	8
$x_i[m]$	0	50	100	100	100	50	0	0
$y_i[m]$	0	0	0	50	100	100	100	50



Fig. 1. Performance parameters for the case of complete communication of state variable matrix  $\mathbf{v}_i$  and a fully connected network. a) Average MDE and b) Average MROE

# A. SSPE performance

We show the performance of the proposed SSPE algorithm in Fig. 1. We consider the case where the exchange of state variables is done without compression. We are aware that without compression the SSPE algorithm is impractical. Nevertheless, we would like to highlight the fact that our algorithm surely converges to the same performance as the centralized DPE approach. Such a characteristic will hopefully motivate more studies in the compression part.

Notice that Fig. 1 shows the average MDE and MROE among the BSs, since all BSs have similar performance. At iteration k=1, all BSs have only local information, that is, there is no prior information. Hence, the estimation is done using only the range likelihood of the received signal which by itself is not enough to estimate the parameters  $\gamma_c$ . As a result, both MDE and MROE are far away from the DPE curves when k=1. At iteration k=2, all BSs exchanged the state variables. In addition, we consider a fully connected network and  $\beta=0.125$  which is the value for fastest convergence<sup>3</sup>. As a consequence, both MDE and MROE converge directly at the second iteration. Lastly, it can also be seen that the SSPE distributely converges to the localization performance of the DPE algorithm.

<sup>3</sup>The value of  $\beta$  for fastest convergence can be obtained based on the eigenvalues of the Laplacian matrix that depends on the logical configuration of the network as explained in [20]



Fig. 2. Performance parameters for the case of compression of state variable matrix  $v_i$  and a fully connected network. a) MDE and b) MROE.

#### B. Compression of state variables

In this subsection we highlight the effect of the compression of the state variables  $\mathbf{v}_i$  over the performance of the proposed algorithm. We show our results in Fig. 2 for the same considerations as in the previous section. Notice that at iteration k=1 the exchange of state variables happens after the estimation of  $\gamma_c$ . Therefore, the performance is exactly the same as the one shown in the previous section; hence they are not shown in the subsequent figures.

It can be seen that the proposed algorithm takes more iterations to achieve convergence even though the value of  $\beta$  for fastest convergence is still used. What is more, it seems that it converges to a value different from the performance of the DPE algorithm. This phenomenon has been studied in the self-synchronization literature under the name of coupling noise effect [17], [16]. In our specific application such coupling noise is introduced to the system by the compression of state variables  $v_i$ . Unlike in [19], [17], where the coupling noise comes from quantization or the communication channel.

On the one hand, for high SNR values ( $\geq$  -9dB), the proposed algorithm converges to the true value. It means that the first two order moments of  $\gamma_c$  describe well enough the state variable  $\mathbf{v}_i$ . Therefore, the compression strategy introduces just a small amount of coupling noise. On the other hand, for medium values of SNR (< -9 dB) the proposed algorithm does not converge to the DPE performance anymore.

It means that the first two order moments of  $\gamma_c$  are not enough to describe completely  $\mathbf{v}_i$ . The reason for this is that, at medium SNR values the range likelihood becomes multimodal. Hence, it is not very well represented with one single Gaussian as we highlighted such phenomenon in [24].

9

We could reduce the coupling noise, and so reducing the bias by two approaches. First, reducing the value of  $\beta$  since according to (18) it multiplies the term that introduces the coupling noise. However, it affects the speed of convergence as it reduces the interaction between nodes. Second, by performing extra iterations without changing the value of  $\beta$ . In summary, both approaches will require more iterations to converge. Hence, the approximated posterior distribution  $\hat{p}$  will get closer and closer to the centralized posterior p with the use of extra iterations.

#### C. IPE - SSPE relationship

It was stated in Section IV that the IPE algorithm can be understood as an special case of the SSPE approach. Such a statement relates the IPE algorithm to the case of a fully connected network. On the one hand, it was empirically shown in [6], [7] that the centralized IPE algorithm converges at iteration k=6. On the other hand and as it will be discussed in Section VI-D, a fully connected network also converges approximately at iteration k=6. Therefore, we consider k=6as last iteration in Fig. 3 to fairly compare the localization error after convergence between the IPE and SSPE approaches. In addition, we also show the curves at iteration k=2, just to show the initial localization error for the algorithms. Notice that we omit the curves at k=1 since the SSPE deviates significantly from the true transmitter position as explained in Section VI-A.

Since both IPE and SSPE exploit the same self-synchronization mechanism, it was expected for both to have similar but not exactly equal MDE and MROE curves. At iteration k = 2, there is a slight difference between the IPE and SSPE algorithms, as can be seen in Fig. 3. However, such difference is only due to the step at which the algorithms compute the final estimate  $\gamma_c$ . For instance, the IPE algorithm computes the final estimates at the FC, i.e., it estimates  $\gamma_c$  after exchanging the state variables. Whereas, the SSPE estimates  $\gamma_c$  before the exchanging. As a result, the IPE algorithm performs in general slightly better than the SSPE algorithm at iteration k=2.

At iteration k=6, both algorithms have converged, and so their performances are very similar to each other. Such minor differences in performance can be explained by the fact that the IPE algorithm only exchanges the pseudo-range information between BSs. Whereas, the SSPE algorithm exchanges more parameters. Such similar performance curves numerically show the relationship between IPE and SSPE algorithms stated in Section IV.

#### D. Convergence Rate Analysis

In this section we show the simulation results of the convergence analysis of the SSPE algorithm described in Section III-D. Such an analysis is a direct consequence of using the self-synchronization mechanism. In the previous



Fig. 3. Comparison between IPE and SSPE algorithms. a) MDE and b) MROE.

section we restricted our analysis to a fully connected network. However our proposed algorithm can work with any network configuration as long as they meet the condition of being strongly connected and balanced. In addition, we study three network configurations that are shown in Fig. 4. Such configurations are shown as undirected graphs since the networks are assumed to be balanced.

As explained in Section III-D, the value of  $\lambda_{max}(\mathbf{W}_k)$  determines the convergence rate of the network. Therefore, Fig. 5 shows the evolution of  $\lambda_{max}(\mathbf{W}_k)$  along the iterations for the three network configurations.

The insights brought by Fig. 5 are twofold. First, it could be seen that the more connections between the BSs the faster the value of  $\lambda_{max}(\mathbf{W}_k)$  goes to zero. Another way to interpret Fig. 5 is the following: the higher the algebraic connectivity the faster the convergence. The fastest convergence is for a fully connected network that is the case studied previously in section VI-A. Second, Fig. 5 can help to determine a threshold value under which we consider that the SSPE algorithm has converged. Consequently, we analyze the number of iterations required for convergence for the considered network configurations to empirically suggest that threshold.

Fig. 6 shows a comparison between the MDE curves for the considered network configurations. Notice that we only analyze the MDE since the behaviour for the MROE is very similar to the MDE. In addition, we decide not to analyze the



10

Fig. 4. Network Configurations: a) 3-regular graph (Wagner graph), b) Irregular graph, c) 2-regular graph (Cycle graph)



Fig. 5. Convergence Rate for different network configurations

Root Mean Squared Error (RMSE) to maintain uniformity on the performance parameters along this paper.

As discussed in Section VI-A, all BSs have the same MDE curve for iteration k=1 (As shown in Fig. 1). Therefore, we do not show them in Fig. 6 since they are the same no matter the network configuration. At iteration k=4, network-a already converges to the DPE performance as shown in Fig. 6(a). Whereas, networks b and c require further iterations. Fig. 6(b) shows the results at iteration k=5 only for network b and c, since network-a already converged at iteration k=4. In addition, Fig. 6(b) also shows the results at iteration k=8 only for network-c, since the other network configurations already converged.

In summary, networks a, b and c need 4, 5, and 8 iterations respectively to converge to the DPE performance. Furthermore, we can see that the values of  $\lambda_{max}(W_k)$ , for such iterations, are under a convergence threshold of  $\lambda_{max}(W_k) \leq 0.15$  (See Fig. 5). Which suggests that it is enough to consider such threshold to achieve convergence on the estimates of the final parameters  $\gamma_c$ . Nevertheless, notice that extra iterations will be needed to reduce the bias due to coupling noise introduced in the system when compressing  $\mathbf{v}_i$ .

Consequently, the number of iterations required for convergence, for a particular network configuration, can be approximately determined as follows.

- Given the network configuration, compute L and choose an acceptable value of β, as studied in [20].
- Compute  $\lambda_{max}(\mathbf{W}_k)$  using (34) for different iterations values.



Fig. 6. MDE comparison for different network configurations at different iterations a) k=4 and b) k=5 and k=8 (Only for network-c).

- Choose the lowest iteration value k for which  $\lambda_{max}(W_k) \leq 0.15$ .
- Add extra iterations to the resulting value to account for the slower convergence due to coupling noise. We found empirically that 4 to 5 extra iterations are enough for the considered network configurations in this paper.

For instance, notice that this could be the reason why the IPE algorithm seems to converge at iteration k=6 empirically suggested in [6]. On the one hand, the IPE can be seen as using the self-synchronization mechanism assuming a fully connected network. On the other hand, a fully connected network converges at the second iteration in the absence of coupling noise, as seen in Fig. 5. Therefore, we could think that the IPE needs 2 iterations plus 4 extra iteration due to coupling noise to reach convergence. Nevertheless, such proposition is still partially based on empirical observations.

#### E. Computational Complexity

We present and discuss the simulation results for the complexity analysis detailed in Section V-B. We discuss the complexity in two ways. We firstly discuss the complexity as a function of the number of iterations. Then, we discuss the complexity as a function of the number of BSs present in the network N.

Fig. 7(a) shows the cumulative number of multiplications for each iteration. We considered the network setup with



11

Fig. 7. Complexity comparison among DPE, IPE and SSPE algorithms per BS/FC a) Complexity as a function of iterations and b) Complexity as a function of BSs in the network.

N=8 BSs as shown in Fig. 4. For the IPE algorithm, the complexity at the BS is less than the complexity at the FC by three orders of magnitude. It means that almost all the IPE complexity is concentrated in the FC. When k=1, the initial SSPE complexity at the BS is higher than the IPE complexity at the FC. However, the complexity increases faster for the IPE than in the SSPE case. When k>5 the resulting SSPE complexity is lower than the IPE complexity. Furthermore, such a SSPE-iteration complexity is even lower for a less connected network (dashed green curve of Fig. 7(a)). Lastly, the sum of SSPE complexities for all the BSs in the network is higher but still comparable to the centralized DPE complexity.

We now discuss the effect of N over the complexity. A dense network with few connections is attractive in the context of distributed algorithms. Therefore, we consider that each BS is directly connected to three other BSs. Fig. 7(b) shows the considered algorithm complexity for k=1 (dashed curves) and k=10 (solid curves). Notice that we omitted the IPE complexity at the BSs, since it is not affected by the number of BSs in the network. When k=1, the complexity at the FC grows with N for the DPE and IPE cases. Conversely, the complexity at the BSs remains the same for the SSPE approach. Furthermore, when N>15, the SSPE has less initial complexity than the IPE approach. At iteration k=10 and when N>17, the IPE complexity is comparable to the DPE case; whereas the SSPE complexity has not increased very much.



Fig. 8. Communication overhead comparison among DPE, IPE and SSPE algorithms

# F. Communication Overhead

In this section we show the simulation results for the analysis presented in Section V-C. We consider N=8 BSs being part of a fully connected network. In addition, we also consider network-c shown in Fig. 4. We present the communication overhead in terms of the received parameters (two parameters per complex value) as a function of the number of iterations as can be seen in Fig. 8.

For the IPE case, the FC receives more parameters than the BSs. In fact, such a difference is approximately one order of magnitude. All the BSs are only connected to the FC and not among each other. As a result, the communication overhead increases only at the FC if more BSs are added to the network.

The SSPE approach spreads the processing among the BSs, thus it increases the necessary communication among them. As a result, the number of received parameters in the BS is higher than the FC or BSs in the IPE approach. Notice that even though the SSPE communication overhead is higher than the IPE case, the SSPE is fully distributed whereas the IPE is centralized. Furthermore, the communication overhead can become comparable to the IPE case for a less connected network as can be seen in the Fig. 8.

# VII. CONCLUSION

In this work, we presented the SSPE algorithm for the case where the transmitter is not time synchronized to the BSs. We also presented a compression approach for the state variables to reduce the amount of communication overhead between the nodes. We also shown the existence of a relationship between IPE and SSPE, since both of them exploit the self-synchronization mechanism. Therefore, the localization is done by means of an iterative process, in which each BS shares just a few parameters between all other BSs, hence the transmitter position is available at each BS at the end of each iteration. Numerical results show that the performance of the final algorithm gets close to the performance of a direct localization in a distributed fashion.

# APPENDIX A RECEIVED SIGNAL LIKELIHOOD MARGINALIZATION

12

We start from an extended version of (13) as

$$p_i(\mathbf{r}_i|a_i,\phi_i,\gamma_c) = C_1 e^{-\frac{1}{\sigma_{w_i}^2}[\mathbf{r}_i - u_i \mathbf{s}_i]^H[\mathbf{r}_i - u_i \mathbf{s}_i]}, \qquad (47)$$

where the  $\gamma_{n_i}$  parameters are fully written. Notice that  $C_1 = (2\pi\sigma_{w_i}^2)^{-P/2}$  is just a normalization factor. Using (10) and (6) and further developing the exponent we have the following expression

$$p_i(\mathbf{r}_i|a_i,\phi_i,\gamma_c) = C_1 e^{-\frac{2}{\sigma_{w_i}^2}\Re(a_i e^{j\phi_i}\mathbf{r}_i^H\mathbf{s}_i) - \frac{1}{\sigma_{w_i}^2}(\mathbf{r}_i^H\mathbf{r}_i + a_i^2\mathbf{s}_i^H\mathbf{s}_i)},$$
(48)

where  $\Re(.)$  is the real part operator. Analyzing the expression, we can make the following three remarks:

- The nuisance parameter  $a_i$  can be outside of the real part operator since it is purely real.
- Only the modulus/amplitude of  $\mathbf{r}_i^H \mathbf{s}_i$  can be outside the real part operator, leaving a phase  $\beta_i = \arg(\mathbf{r}_i^H \mathbf{s}_i)$ .
- The term  $\mathbf{r}_i^H \mathbf{r}_i$  is constant; hence it can be gathered with the normalization factor.

Applying the remarks, (48) can be expressed as

$$p_i(\mathbf{r}_i|a_i,\phi_i,\gamma_c) = C_2 e^{-\frac{2a_i}{\sigma_{w_i}^2}|\mathbf{r}_i^H \mathbf{s}_i|\cos\left(\phi_i + \beta_i\right) - \frac{1}{\sigma_{w_i}^2}(a_i^2 \mathbf{s}_i^H \mathbf{s}_i)},$$
(49)

where  $C_2$  is a constant taking into account  $C_1$  and the exponential constant term of  $\mathbf{r}_i^H \mathbf{r}_i$ .

Simultaneously, we can write the extended version of (14) as

$$p_i(\mathbf{r}_i|\gamma_c) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_i(\mathbf{r}_i|a_i,\phi_i,\gamma_c) p(a_i) p(\phi_i) da_i d\phi_i$$
(50)

where replacing (49) in (50), rearranging the terms and the order of integration as

$$p_i(\mathbf{r}_i|\gamma_c) = C_3 \int_{-\infty}^{\infty} G e^{\frac{-a_i^2}{\sigma_{w_i}^2}(\mathbf{s}_i^H \mathbf{s}_i)} p(a_i) da_i$$
(51)

where  $C_3 = C_2 p(\phi_i)$ , since  $\phi_i$  is uniformly distributed. The factor G is defined as the following integral

$$G = \int_{-\pi}^{\pi} e^{jz\cos\left(\phi_i + \beta_i\right)} d\phi_i \tag{52}$$

where  $z = -j \frac{2a_i}{\sigma_{w_i}^2} |\mathbf{r}_i^H \mathbf{s}_i|$ . Since  $\beta_i$  is a constant phase we can perform a change of variable as:

$$G = \int_{-\pi+\beta_i}^{\pi+\beta_i} e^{jz\cos\left(\theta_i\right)} d\theta_i = 2 \int_0^{\pi} e^{jz\cos\left(\theta_i\right)} d\theta_i \qquad (53)$$

where  $\theta_i = \phi_i + \beta_i$ . Note that the integral is periodic regarding  $\theta_i$ ; hence the integration interval remains to be  $[-\pi, \pi]$ . In addition, the integral is symmetric regarding the origin; thus, it is sufficient to consider the integration interval  $[0, \pi]$ . Finally, the last integral in (53) is the integral form of the Bessel function of the first kind of order zero  $J_0(z)$ as defined in [25]. Therefore  $G = 2\pi J_0(z)$  can be replaced in (51) to finally obtain the desired expression shown in (15).

# APPENDIX B

# ITERATIVE POSITIONING ESTIMATION

# Step 1 - Intermediate Parameters Estimation

At BS-*i* the log-posterior can be expressed as:

$$\mathcal{P}_i(\psi_i|\mathbf{r}_i)[k+1] = \mathcal{L}_i(\mathbf{r}_i|\psi_i) + \mathcal{P}_i(\psi_i)[k]$$
(54)

where  $\psi_i$  is a vector of intermediate parameters that are function of the common parameters  $\gamma_c$ , i.e.,  $\psi_i = \psi_i(\gamma_c)$ . The term  $\mathcal{P}_i(\psi_i)$  is a log-prior distribution of such intermediate parameters that is updated at the FC in step 2. For instance, in [6] only one intermediate parameter was used  $\psi_i = d_i(x, y)$ with  $\gamma_c = [x, y]^T$ . The first two order moments of  $\psi_i$  are sent to the FC, i.e.,  $(\hat{\psi}_i, \sigma_{\hat{\psi}_i}^2)$  which are computed using the LOTUS rule shown in (23) with respect to  $p_i(\psi_i | \mathbf{r}_i)$  at iteration k.

# Step 2 - Prior Estimation

The prior is refined at the FC using the information sent by all BSs, as

$$\mathcal{P}_{i}(\gamma_{c}|\hat{\psi}_{1}...\hat{\psi}_{N}\setminus\hat{\psi}_{i}) = \sum_{j\neq i}^{N} -\frac{1}{2\sigma_{\hat{\psi}_{j}}^{2}}|\psi_{j}-\hat{\psi}_{j}|_{2}^{2} +\log(C_{\mathcal{P}_{i}}),$$
(55)

where  $C_{\mathcal{P}_i}$  is a normalization factor. Since  $\psi_i$  depend on  $\gamma_c$ , we can compute the first two order moments of  $\psi_i$  using the LOTUS rule regarding  $\mathcal{P}_i(\gamma_c | \hat{\psi}_1 ... \hat{\psi}_N \setminus \hat{\psi}_i)$ . Such parameters  $(\hat{\psi}_i, \sigma_{\hat{\psi}_i}^2)$  are sent to the BS-*i* to update the prior distribution in (54).

To obtain (36), we subtract  $\mathcal{P}_i(\psi_i | \mathbf{r}_i)[k]$  from both sides leading to the expression:

$$\Delta \mathcal{P}_i(\psi_i | \mathbf{r}_i)[k] = \mathcal{L}_i(\mathbf{r}_i | \psi_i) + \mathcal{P}_i(\psi_i)[k] - \mathcal{P}_i(\psi_i | \mathbf{r}_i)[k],$$
(56)

where  $\Delta \mathcal{P}_i[k] = \mathcal{P}_i[k+1] - \mathcal{P}_i[k]$ . Finally, without changing the IPE iterative framework, we consider the case where  $\psi_i = \gamma_c$ , i.e., we consider that the intermediate parameters are the final estimated parameters. Under such assumption we can change the dependency of (56) to obtain the expression shown in (36).

#### ACKNOWLEDGMENT

The authors acknowledge the financial support of the Walloon Region through the WIN2Wal/2018/1/DI/34 LUMINET project.

#### REFERENCES

- "Evolved universal terrestrial radio access (e-utra); physical channels and modulation (release 13)," in *Technical Specification 36.211 V13.13.0*, January 2020.
- [2] P. Closas, C. Fernandez-Prades, and J. A. Fernandez-Rubio, "Maximum likelihood estimation of position in gnss," *IEEE Signal Processing Letters*, vol. 14, no. 5, pp. 359–362, 2007.
- [3] A. J. Weiss, "Direct position determination of narrowband radio frequency transmitters," *IEEE Signal Processing Letters*, vol. 11, no. 5, pp. 513–516, 2004.
- [4] R. Zekavat and R. M. Buehrer, "Source localization: Algorithms and analysis," in *Handbook of Position Location: Theory, Practice, and Advances*, 2019, pp. 59–106.
- [5] A. Amar and A. J. Weiss, "New asymptotic results on two fundamental approaches to mobile terminal location," in 2008 3rd International Symposium on Communications, Control and Signal Processing, 2008, pp. 1320–1323.

- [6] F. Horlin, M. Van Eeckhaute, T. Van der Vorst, A. Bourdoux, F. Quitin, and P. De Doncker, "Iterative toa-based terminal positioning in emerging cellular systems," in 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–5.
- [7] M. Van Eeckhaute, T. Van der Vorst, A. Bourdoux, F. Quitin, P. De Doncker, and F. Horlin, "Low complexity iterative localization of time-misaligned terminals in cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10730–10739, 2018.
- [8] M. Van Eeckhaute, T. Van der Vorst, F. Quitin, P. De Doncker, and F. Horlin, "Toa-based iterative localization in rich multipath channels," in 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2018, pp. 1–5.
- [9] S. Monfared, A. Delepaut, M. Van Eeckhaute, P. De Doncker, and F. Horlin, "Iterative localization method using aoa for iot sensor networks," in 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), 2019, pp. 1–6.
- [10] M. Eeckhaute, E. Gottlob, P. Doncker, and F. Horlin, "Iterative rtof-based localization and time synchronization in wlan-like systems," in *Electronics Letters*, vol. 54, no. 24, 10 2018, pp. 1386–1387.
- [11] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links—part i: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008.
- [12] W. Yu, N. D. Gaubitch, and R. Heusdens, "Distributed tdoa-based indoor source localisation," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 6887–6891.
- [13] A. Simonetto and G. Leus, "Distributed maximum likelihood sensor network localization," *IEEE Transactions on Signal Processing*, vol. 62, no. 6, pp. 1424–1437, 2014.
- [14] W. Cui, S. Wu, Y. Wang, and Y. Shan, "A gossip-based tdoa distributed localization algorithm for wireless sensor networks," in 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013, pp. 783–788.
- [15] G. Soatti, M. Nicoli, A. Matera, S. Schiaroli, and U. Spagnolini, "Weighted consensus algorithms for distributed localization in cooperative wireless networks," in 2014 11th International Symposium on Wireless Communications Systems (ISWCS), 2014, pp. 116–120.
- [16] S. Barbarossa and G. Scutari, "Bio-inspired sensor network design," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26–35, 2007.
- [17] A. Fasano and G. Scutari, "The effect of additive noise on consensus achievement in wireless sensor networks," in 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 2277–2280.
- [18] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, 2009.
- [19] H. Song, L. Yu, and W. Zhang, "Quantized consensus in noisy sensor networks," in *Proceedings of the 30th Chinese Control Conference*, 2011, pp. 4994–4998.
- [20] S. Dhuli, K. Gaurav, and Y. N. Singh, "Convergence analysis for regular wireless consensus networks," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4522–4531, 2015.
- [21] G. Scutari, S. Barbarossa, and L. Pescosolido, "Distributed decision through self-synchronizing sensor networks in the presence of propagation delays and asymmetric channels," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1667–1684, 2008.
- [22] L. Chen, G. Carpenter, S. Greenberg, J. Frolik, and X. S. Wang, "An implementation of decentralized consensus building in sensor networks," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 667–675, 2011.
- [23] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973. [Online]. Available: http://dml.cz/dmlcz/101168
- [24] E. Pocoma, F. Rottenberg, F. Quitin, L. Vandendorpe, P. De Doncker, and F. Horlin, "Iterative toa-based localization of wireless transmitters using dirichlet-kernel-based range representation," in 2021 IEEE Vehicular Technology Conference (VTC), 2021, pp. 1–5.
- [25] F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, *NIST Handbook of Mathematical Functions*, 1st ed. USA: Cambridge University Press, 2010.

IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. XX, NO. X, FEBRUARY 2022

**Evert I. Pocoma Copa** was born in 1990 in El Alto, Bolivia. He received a scholarship from the Foundation Simón I. Patiño that helped him to receive the M.Sc. degree in electrical engineering from the Université Libre de Bruxelles (ULB), Brussels, Belgium, in 2019. Since September 2019, he is working toward the Ph.D degree with OPERA WCG, ULB. He is currently working on distributed localization and tracking techniques in wireless sensor networks.



**François Horlin** received the Ph.D. degree from the Université Catholique de Louvain, Louvain-La-Neuve, Belgium, in 2002. He specialized in the field of signal processing for digital communications. He joined the Interuniversity Micro-Electronics Center in 2002. He led the project aiming at developing a 4G cellular communication system in collaboration with Samsung Korea. In 2007, he became a Professor with the Université Libre de Bruxelles (ULB), Brussels, Belgium. He is currently supervising

a research team working on next-generation communication systems. Localization based on 5G signals, filterbank-based modulations, massive multiple input and multiple output, and dynamic spectrum access are examples of currently investigated research topics. He has been an Academic Representative to the Executive Board of ULB from 2010 to 2015. Since 2017, he has been the Vice Dean for research with the Ecole Polytechnique de Bruxelles.



François Quitin received the Ph.D. degree in electrical engineering from the Université Libre de Bruxelles (ULB), Brussels, Belgium, and the Université Catholique de Louvain, Louvain-La-Neuve, Belgium, in 2011. He is an Assistant Professor with ULB. From 2011 to 2013, he was a Postdoctoral Researcher with the University of California, Santa Barbara, and from 2013 to 2015, he was a Postdoctoral Research Fellow with Nanyang Technological University, Singapore. His research interests include

experimental and prototyping aspects in wireless communications, taking advanced theoretical ideas all the way to practice. He was the recipient of the Alcatel-Lucent Bell Scientific Award in 2012.



Luc Vandendorpe (Fellow, IEEE) was born in Mouscron, Belgium, in 1962. He received the degree (summa cum laude) in electrical engineering and the Ph.D. degree in applied science from the Catholic University of Louvain (UCLouvain), Louvain La Neuve, Belgium, in 1985 and 1991, respectively.,Since 1985, he has been with the Communications and Remote Sensing Laboratory, UCL, where he first worked in the field of bit rate reduction techniques for video coding. In 1992,

he was a Visiting Scientist and a Research Fellow with the Telecommunications and Traffic Control Systems Group, Delft Technical University, The Netherlands, where he worked on spread spectrum techniques for personal communications systems. From October 1992 to August 1997, he was a Senior Research Associate with the Belgian NSF, UCL, and an invited Assistant Professor. He is currently a Full Professor with the Institute for Information and Communication Technologies, Electronics, and Applied Mathematics, UCLouvain. His research interests include digital communication systems and more precisely resource allocation for OFDM(A)-based multicell systems, MIMO and distributed MIMO, sensor networks, UWB-based positioning, and wireless power transfer. He was the Chair of the IEEE Benelux Joint Chapter on Communications and Vehicular Technology from 1999 to 2003. He was a Co-Technical Chair for IEEE ICASSP 2006. He served as an Editor for Synchronization and Equalization of IEEE Transactions on Communications from 2000 to 2002, and as an Associate Editor for IEEE Transactions on Wireless Communications from 2003 to 2005, and IEEE Transactions on Signal Processing from 2004 to 2006. He is or has been a TPC Member for numerous IEEE conferences (VTC, GLOBECOM, SPAWC, ICC, PIMRC, and WCNC). He was an Elected Member of the Signal Processing for Communications Committee from 2000 to 2005, and the Sensor Array and Multichannel Signal Processing Committee of the Signal Processing Society from 2006 to 2008 and from 2009 to 2011.



**Philippe De Doncker** received the M.Sc. degree in physics engineering and the Ph.D. degree in science engineering from the Université Libre de Bruxelles (ULB), Brussels, Belgium, in 1996 and 2001, respectively. He is currently a Professor with the ULB, where he leads the research activities on wireless channel modeling and (bio) electromagnetics. 14