Joint Communication and Sensing in OTFS-based UAV Networks

Joint Communication and Sensing in OTFS-based UAV Networks

Alessandro Nordio, *Member, IEEE,* Carla Fabiana Chiasserini, *Fellow, IEEE,* Emanuele Viterbo, *Fellow, IEEE*

***Abstract*—We consider the problem of accurately localizing N unmanned aerial vehicles (UAV) in 3D space where the UAVs are part of a swarm and communicate with each other through orthogonal time-frequency space (OTFS) modulated signals. The OTFS communication system operates in the delay-Doppler domain and can simultaneously provide range and velocity information about the scatterers in the channels at no additional cost. Each receiving UAV estimates the multipath wireless channel on each link formed by the line-of-sight (LoS) transmission and by the single reflections from the remaining $N-2$ UAVs. The estimated channel delay profiles are communicated to an edge server to estimate the location and velocity of the UAVs from the relative echo delay (RED) measurements between the LoS and the non-LoS paths. To accurately obtain such estimations, we propose a solution called Turbo Iterative Positioning (TIP), initialized by a belief-propagation approach. Enabling a full cold start (no prior knowledge of initial positions), the belief propagation first provides a map associating each echo to a reflecting UAV. The localization of the N UAVs is then derived by iteratively alternating a gradient descent optimization and a refinement of the association maps between UAVs and echos. Given that the OTFS receivers also acquire the Doppler shifts of each path, the UAV's velocities can be sensed jointly with communication. Our numerical results, obtained also using real-world traces, show how the multipath links are beneficial to achieving very accurate position and velocity for all UAVs, even with a limited delay-Doppler resolution. The robustness of our scheme is proven by its performance approaching the Cramér-Rao bound.**

***Index Terms*—UAVs, OTFS, Mobile networks, Position, Velocity, Localization**

I. Introduction

Accurate localization stands as a crucial element in the repertoire of applications that forthcoming 6G communication networks are anticipated to empower [2]. This encompasses applications such as ensuring safety in vehicular networks and facilitating area exploration, rescue operations, and relief efforts, all of which can be achieved through swarms of unmanned aerial vehicles (UAVs) [3], [4]. In situations where GPS signals are restricted or absent, the process of localizing these communicating nodes typically involves measuring their distances, which is performed by estimating the time delay of pilot signals' propagation.

Alessandro Nordio is with CNR-IEIIT, Torino, Italy; Carla Fabiana Chiasserini is with Politecnico di Torino, Torino, Italy; Emanuele Viterbo is with Monash University, Melbourne, Australia.

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART"), and by the Project: "SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics". E. Viterbo's work is supported by the Australian Research Council (ARC) through the Discovery project: DP200100096.

A preliminary version of this work was presented at Globecom'23 [1].

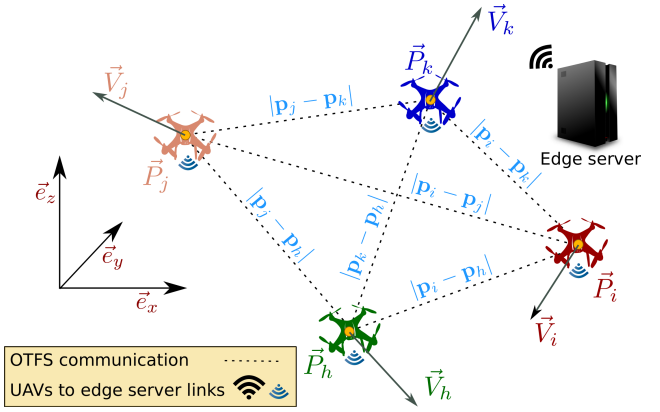

Fig. 1. Communicating UAVs assisted by an edge server. The UAV u ($u \in \{h, i, j, k\}$) has position $\vec{P}_u$ and velocity $\vec{V}_u$, measured with respect to a common Cartesian coordinate system. The notation $|\mathbf{p}_u - \mathbf{p}_v|$ denotes the distance between UAV u and UAV v, where $\mathbf{p}_u$ and $\mathbf{p}_v$ are the components of the geometric vectors $\vec{P}_u$ and $\vec{P}_v$, respectively.

In such demanding scenarios, it is important to integrate the communication system with a multi-target localization system in the most efficient way to provide savings in bandwidth, power, and hardware resources. This can be achieved by sharing a single waveform for both sensing and communications by leveraging the pilot-based channel estimation of the communication system to acquire the range and velocity information.

The integration of communication and sensing in a network environment can offer further gains but also presents additional challenges when considering synchronization requirements across all nodes. Specifically, to effectively combine the range and velocity information collected from the nodes in the network, it is typically assumed that they all have a very accurate timing reference, down to carrier and/or symbol synchronization level. However, this is not realistic in most plesiochronous communication networks.

In this work, we focus on a joint *network* communication and sensing scenario (shown in Fig. 1) involving a UAV swarm, where nodes communicate with each other as well as with an edge server via orthogonal time frequency space (OTFS) signaling. OTFS-based communication takes advantage of the sparse multipath representation of the channels in the delay-Doppler domain, which can also be effectively used to acquire information about the range and velocity of the scatterers in the scene [5].

Importantly, our solution exploits the delay-Doppler domain channel estimation used by the OTFS communication system to achieve precise positions and velocities of all UAVs within the swarm. It is worth noting that, by operating in the delay-

Doppler domain, OTFS can accurately resolve all the multiple paths with distinct delays and Doppler shifts. Indeed, although the path delay resolution used for position estimation is the same for both OTFS and orthogonal frequency-division multiplexing (OFDM) (being inversely proportional to the waveform bandwidth), OFDM cannot resolve paths with the same delay and distinct Doppler shifts, as it operates in the time-frequency domain. It follows that using OFDM results in large errors in the velocity estimation [6].

Furthermore, unlike most of the positioning algorithms available in the literature, our approach is designed to use the relative echo delay (RED) measurements between line-of-sight (LoS) and non-LoS (nLoS) paths for each OTFS communication link. The main advantage of using RED profiles over the more commonly assumed time-of-arrival (ToA) method is that, by exploiting nLoS paths, it enables localization without the need for tight synchronization among the nodes. The RED profile between LoS and nLoS paths is extracted from the delay domain enabling the UAVs to operate without the need for fine-grained, sample-level synchronization (within the inverse of the communication bandwidth). Specifically, the RED and Doppler shift of each path obtained from the delay-Doppler domain channel estimate enable us to derive the corresponding UAVs' positions and velocities. Even more importantly, RED enables us to take full advantage of the multipath channels in the network, which usually hurt the classic position and velocity estimation techniques.

Although the limited communication channel bandwidth yields low-resolution estimates of the REDs in the multipath channels between UAV pairs, this is tempered by the availability at the edge server of the full multipath channel measurements. The edge server first attempts to associate the RED measurements with the paths identified by the reflecting UAVs, i.e., to solve what in the localization literature is known as the *assignment problem.* The edge server subsequently uses this information to estimate the UAVs' positions, by applying a gradient descent algorithm that converges to the actual values of the UAVs' positions. As the last step, the UAVs' velocities can be derived from the Doppler shift measurements.

To jointly solve our specific assignment problem for all the RED profiles collected at the edge server, we employ a belief propagation (BP) algorithm that exploits the redundant RED measurements at all UAVs and their interdependence. Then, we propose a low complexity *turbo-iterative positioning* (TIP) algorithm that iteratively refines the position estimates by correcting residual errors in the initial solution to the assignment problem.

To evaluate the performance of our approach, we resort to comparisons to the Cramér Rao bounds on the estimation errors. Moreover, by leveraging real-world traces, we demonstrate that TIP (i) yields highly accurate estimated positions and velocities, and (ii) is resilient to noise, approaching the Cramér-Rao bound.

To summarize, our main contributions are as follows:

- We define the joint network communication and sensing scenario for a swarm of UAVs and propose the use of OTFS communications pilots for delay-Doppler channel estimation to acquire the position and velocity of all UAVs in the swarm from the RED profiles;
- We introduce a novel solution to the initial network RED assignment problem leveraging belief propagation. Importantly, our approach innovates beyond the existing literature where rarely multipath propagation has been considered to help in classic positioning techniques;
- We design a new low-complexity iterative algorithm (TIP) to refine the initial assignment estimates and improve the accuracy of positions and velocities;
- We demonstrate the effectiveness of our solution framework through numerical results and real-world UAV mobility traces, relative to the Cramér-Rao bound. Importantly, we show that our solution substantially enhances the positioning accuracy beyond the limits of the communication bandwidth constraints.

The rest of the paper is organized as follows. We introduce the notation used in this work in Sec. III. Then, after detailing the assumptions and the problem formulation, Sec. IV provides a roadmap to our solution. The fundamental blocks of the proposed algorithm are presented in Secs. V–VII, with Sec. V describing how to map the channel delay profiles to the UAV identities using BP, and Sec. VI introducing the proposed estimators for the UAV's positions and velocities. Sec. VII combines the previous results in a highly efficient positioning algorithm, and Sec. VIII presents the joint Cramér Rao lower bound on the variance of the position-velocity estimator. Finally, we analyze the performance of our algorithm in Sec. IX. Conclusions and future research directions are discussed in Sec. X.

II. Related Work

Despite the extensive research on using OFDM waveforms for this purpose [7], [8], the challenge of achieving precise localization in nLoS and high-mobility scenarios persists [9]. To address this, OTFS has been explored as an alternative to OFDM. In particular, [10] demonstrates the effectiveness of OTFS-modulated physical random access channel (PRACH) transmissions for ToA based ranging, while [9] designs an OFTS transceiver for transmitting and receiving positioning reference signals and using ToA for belief propagation-based cooperative positioning.

In cooperative positioning techniques, a number of nodes (called agents) exchange information to enhance the overall accuracy of range and velocity estimation for all the nodes in the network [11], [12]. A large body of work foresees agents that combine information from multiple sensors such as cameras, LiDAR, GPS and inertial measurements units (IMU) to improve localization accuracy, see, e.g., [13], [14], [15], [16]. Alternatively, a localization approach based on antenna arrays and angle of arrival estimation of multipath signals is proposed in [17]. A vast literature is available on multi-agent localization and tracking, for which we refer the interested reader to [18]. For example, the problem of localizing multiple targets in a multipath environment using a set of distributed single-antenna receivers is addressed in [19] and solved using a Bayesian estimation technique. The same problem is considered in [20] using, as observations, a sequence of video frames. Alternative approaches to multitarget

tracking in clutter use probabilistic data association techniques [21], [22]. We underline that all these localization techniques require additional sensing equipment or antennas. Instead, at the essence of joint communication and sensing is realizing localization through the exclusive use of the communication system without spending additional resources [23], [24].

Given the observations collected by the agents, the problem of determining the positions of the targets from a set of measurements can be viewed as a least square error estimation problem, where the association of the measurements to the variables is unknown. Jointly solving these two problems to the optimum has a prohibitive complexity. Nevertheless, high-performance multi-target tracking solutions based on message passing belief propagation (BP) algorithms have proven effective for their low complexity and flexibility [25]. For instance, in [26] BP is applied to a simultaneous localization and mapping problem, in a multipath propagation environment where paths are associated to virtual anchors in a probabilistic fashion.

To the best of our knowledge, there are no existing works leveraging RED profiles. The RED profile technique is part of our novel contribution, and we have coined such terminology. The RED profile technique constructively exploits the multipath nature of the channels and avoids the need for precise synchronization among UAVs. Typical solutions rely on the LoS paths only and assume precise synchronization. Indeed, it is worth noting that our approach differs from TDoA-based positioning with multiple sensor nodes, where only LoS paths are considered and signal cross-correlations need to be evaluated. Additionally, TDoA still requires either strong synchronization among the sensors [27], or complex compensation techniques [28].

Finally, we mention that an early version of this work has appeared in our conference paper [1]. With respect to [1], we now integrate both positions and velocities estimation, and present a substantially improved version of the envisioned methodology as well as additional performance results.

III. Notations

We denote *geometric vectors* by capital letters with an arrow on top, such as $\vec{A}$. Boldface uppercase and lowercase letters denote matrices and vectors, respectively. The transpose of matrix $\mathbf{A}$ is denoted by $\mathbf{A}^{\mathsf{T}}$, whereas $[\mathbf{A}]_{i,j}$ indicates its (i,j)-th element. $\mathbf{I}$ is the identity matrix and the L2-norm of the vector $\mathbf{a}$ is represented by $|\mathbf{a}|$. Given a Cartesian coordinate system with basis vectors $\vec{e}_x$, $\vec{e}_y$, and $\vec{e}_z$, the geometric vector $\vec{A}$ is represented as a three-dimensional column vector $\mathbf{a} = [\langle\vec{A},\vec{e}_x\rangle, \langle\vec{A},\vec{e}_y\rangle, \langle\vec{A},\vec{e}_z\rangle]^{\mathsf{T}}$. Sets, lists, or maps are denoted by calligraphic or Greek capital letters. The notation $[\cdot]_{\neq}$ means that all elements of the list $[\cdot]$ have different values. The estimate of the quantity a is denoted by $\widehat{a}$. The probability of an event A is referred to as $\mathbb{P}(A)$, while the probability density function (pdf) of the random variable a is denoted by $f_a(a)$. The symbol $\mathbb{E}_a[\cdot]$ stands for expectation with respect to the random variable a. Finally, the Gaussian distribution with zero mean and variance σ^2 is denoted by $\mathcal{N}(0,\sigma^2)$ and the uniform distribution between a and b is denoted by $\mathcal{U}[a,b]$.

The main variables and parameters used throughout the paper are summarized in Table I.

IV. System Model and Problem Statement

The scenario under study includes a swarm of N UAVs located in a 3D flight area. The position and the velocity of the generic UAV i (i=1,…,N) are denoted by the geometric vectors $\vec{P}_i$ and $\vec{V}_i$, respectively, which refer to a common Cartesian coordinate system. We also use the symbol $\mathbf{p}_i$=$[p_{i,x}, p_{i,y}, p_{i,z}]^{\mathsf{T}}$ to denote the components of $\vec{P}_i$ along the x, y, and z axis of the coordinate system and, similarly, $\mathbf{v}_i$=$[v_{i,x}, v_{i,y}, v_{i,z}]^{\mathsf{T}}$ to denote the components of the velocity vector $\vec{V}_i$.

A subset of A UAVs act as *anchors* for the system, i.e., their positions and velocities are perfectly known while the positions and velocities of the remaining $\bar{N}$=$N-A$ UAVs are unknown and have to be estimated. The role of the anchors is to resolve translation and rotation ambiguities when estimating the position and velocity of the UAVs. The use of anchors is commonly considered in the literature, e.g., in [18], [26] and their position is known to the edge server. In a 3D space, at least 4 non-coplanar anchors are required. With only 3 anchors, the estimation problem becomes geometrically ill-conditioned and may result in invalid solutions. Since 3 anchors lie on the same plane, they are not sufficient to determine on which side of the plane a UAV is located.

As depicted in Fig. 1, the UAVs communicate with each other, as well as with an edge server controlling the geographical area of interest, using OTFS waveforms. The wireless channel connecting any two UAVs presents high-mobility multipath fading characteristics. For simplicity, we assume that no physical obstacle, other than the UAVs belonging to the swarm, can reflect or block the signal. Thus, each communication channel between any pair of UAVs is characterized by the LoS path and $N-2$ nLoS paths due to a single reflection from the other UAVs in the swarm. Then the time-delay expression of the channel connecting UAVs i and j ($j\neq i$) is given by [5]:

$$h_{i,j}(t,\tau) = \sum_{k=1,k\neq i}^{N} c_{i,j,k} \mathrm{e}^{\mathrm{j}2\pi\nu_{i,j,k}t}\delta(\tau-\tau_{i,j,k}) \qquad (1)$$

where $c_{i,j,k}$, $\tau_{i,j,k}$, and $\nu_{i,j,k}$ are, respectively, the channel coefficients, the path delay, and the path Doppler shifts generated by the signal reflection on the k-th UAV. When k=j, we have the *LoS* path, otherwise we have the *nLoS* paths. We assume that the pilot power is sufficient for the channel estimation units of all OTFS receivers to identify all the channel parameters in (1).

In a practical scenario, the UAV receivers are not synchronized with respect to a common time reference. Leveraging on the delay-Doppler domain channel estimation of OTFS, the RED profiles of each communication link between UAVs i and j can be measured, locally at each receiver. This is obtained by using the delays of the echos from all the nLoS paths relative to the LoS path (zero relative delay). Then, as shown in Fig. 1, the RED $\tau_{i,j,k}$ corresponds to the relative delay between the k-th nLoS path and the LoS path, and it is given by

TABLE I
VARIABLES AND PARAMETERS

Symbol	Description
N	Total number of UAVs
A	Number of anchor UAVs
f_c, B	Signal central frequency and bandwidth
T_f	OTFS frame duration
M_d, N_D	Number of delay and Doppler bins
$\mathbf{p}_i$, $\mathbf{v}_i$	Components of the position and velocity vectors of the i-th UAV
$\tau_{i,j,k}$	RED between the k-th non-LoS path and the LoS path, when UAV j communicates with UAV i
$\delta_{i,j,k}$	Distance corresponding to the propagation delay $\tau_{i,j,k}$
$\widetilde{\delta}_{i,j,k}$	Discretized distance $\delta_{i,j,k}$
$\nu_{i,j,k}$	Doppler shift of the signal transmitted by UAV j, reflected by UAV k, and received at UAV i
$\omega_{i,j,k}$	Velocity corresponding to the Doppler shift $\nu_{i,j,k}$
$\widetilde{\omega}_{i,j,k}$	Discretized velocity $\omega_{i,j,k}$
$\mathcal{D}_{i,j}$, $\mathcal{V}_{i,j}$	Lists describing the profile of the channel connecting UAV i and UAV j
$\mu_{i,j}$	Map associating the elements of $\mathcal{D}_{i,j}$ and $\mathcal{V}_{i,j}$ to the UAV's identities

$$\tau_{i,j,k} = \tau_{j,k} + \tau_{k,i} - \tau_{i,j} \tag{2}$$

where $\tau_{u,v}{=}\frac{1}{c}|\mathbf{p}_u - \mathbf{p}_v|$, $u, v{\in}\{i,j,k\}$, is the absolute signal propagation delay from UAV u to UAV v, and c is the speed of light. Clearly, for the LoS path, $\tau_{i,j,j}{=}0$ by definition.

The REDs $\tau_{i,j,k}$ depend upon the swarm geometry and, for each one of them, we define the corresponding distance[1]

$$\delta_{i,j,k} \triangleq c\,\tau_{i,j,k} = |\mathbf{p}_j - \mathbf{p}_k| + |\mathbf{p}_k - \mathbf{p}_i| - |\mathbf{p}_i - \mathbf{p}_j|\,. \tag{3}$$

Notice also that, due to channel reciprocity, $\delta_{i,j,k}{=}\delta_{j,i,k}$ $\forall i,j$, and $k{\neq}\{i,j\}$. The delays $\tau_{i,j,k}$ or, equivalently, the distances in (3) can be measured through pilot-based OTFS channel estimation techniques to a degree of accuracy dependent upon the signal bandwidth B [29].

The Doppler shifts $\nu_{i,j,k}$ are instead functions of both UAV's velocities and positions. For each of them, we define the corresponding *radial velocity*[2], $\omega_{i,j,k}{\triangleq}\frac{c}{f_c}\nu_{i,j,k}$, which as shown in Appendix A, can be written as:

$$\omega_{i,j,k} = (\mathbf{v}_j - \mathbf{v}_k)^{\mathsf{T}}\mathbf{u}_{j,k}(\mathbf{p}) + (\mathbf{v}_k - \mathbf{v}_i)^{\mathsf{T}}\mathbf{u}_{k,i}(\mathbf{p}) \tag{4}$$

where $\mathbf{p}{=}[\mathbf{p}_1^{\mathsf{T}}, \ldots, \mathbf{p}_N^{\mathsf{T}}]^{\mathsf{T}}$ is a column vector of size $3N$ stacking all components of the position vectors $\vec{P}_i$ ($i{=}1,\ldots,N$), f_c is the carrier frequency and the elements of

$$\mathbf{u}_{u,v}(\mathbf{p}) = \frac{\mathbf{p}_u - \mathbf{p}_v}{|\mathbf{p}_u - \mathbf{p}_v|} \tag{5}$$

are the components of the versor pointing from UAV v to UAV u.

[1]We refer to this as the distance for convenience; more precisely, it is the difference in length between nLos and LoS paths.

[2]The velocity $\omega_{i,j,k}$ is the radial component of the velocity vector $\vec{V}_j$, measured by the i-th receiving UAV through the reflected path by the k-th UAV.

A. OTFS communication and channel estimation

OTFS is a communication scheme suitable for high-mobility communication environments (like the one considered in this work) exhibiting channels with multiple paths with different delays and Doppler shifts. The goal of our paper is to demonstrate that accurate sensing of the position and velocity of highly mobile nodes in a network is possible by leveraging the OTFS scheme. Given the large diversity gains offered by OTFS, a reasonable SNR – far below the one offered by conventional OFDM – is sufficient to achieve a target error performance. Since the focus of this work is on localization and tracking, we do not report the performance of the communication links for the sake of brevity. However, for interested readers, this can be found in the vast literature demonstrating the superiority of OTFS over OFDM in a high-mobility scenario [5].

In OTFS, the delay Doppler (DD) domain is discretized into an $M_d{\times}N_D$ grid resulting in a delay resolution of $\Delta\tau{=}\frac{T}{M_d}{=}\frac{1}{B}$ and a Doppler shift resolution of $\Delta\nu{=}\frac{1}{N_D T}$, where $T{=}\frac{1}{\Delta f}$ is the duration of a block in a frame of duration $T_f{=}N_D T$ and $B{=}M_d\Delta f$ is the communication channel bandwidth [5]. Simple channel estimation techniques, such as the one in [29], enable estimating the delay and Doppler shift of each path with the above resolution. Pilot symbols can be embedded in the delay-Doppler together with the data, as shown in [5, Ch. 7]. Such pilots are spread across the entire transmission frame ($T_f{=}NT$) and bandwidth ($B{=}M\Delta f{=}M/T$), timely capturing all the channel information relevant for the frame detection. By operating within T_f seconds, we avoid the problem of channel information aging.

By partitioning the DD domain grid into disjoint rectangular tiles, UAVs can simultaneously broadcast a pilot in a single frame along with data, thus allowing all the receivers to perform channel estimation simultaneously within T_f seconds [5]. This scheme avoids pilot interference among transmitting UAVs.

Here we assume that the multipath channel exhibits fractional delays and Doppler shifts, and that, for each delay, there is only one Doppler shift. In turn, this allows estimating the distances $\delta_{i,j,k}$ and the velocities $\omega_{i,j,k}$ from the measurements

$$\widetilde{\delta}_{i,j,k} = \delta_{i,j,k} + \eta_{i,j,k} \tag{6}$$

$$\widetilde{\omega}_{i,j,k} = \omega_{i,j,k} + \zeta_{i,j,k} \tag{7}$$

where $\eta_{i,j,k}$ and $\zeta_{i,j,k}$ are the errors affecting, respectively, distance and velocity estimation. These errors are due to the quantization noise caused by the limited DD resolutions $\Delta\tau$ and $\Delta\nu$ as well as the effect of thermal noise. While the thermal noise component contributing to $\eta_{i,j,k}$ and $\zeta_{i,j,k}$ can be reduced by increasing the pilot power [29], the quantization noise can only be reduced at the greater cost of increasing the bandwidth B and the frame length T_f.

In addition to quantization noise some other causes of degradation in the delay-Doppler estimates may be considered. The motion of the UAV propellers may introduce spurious contributions to the channel delay-Doppler profile. Also, despite the frame time is typically a small fraction of a second (e.g., 20 ms), sudden UAV accelerations may degrade the estimate of

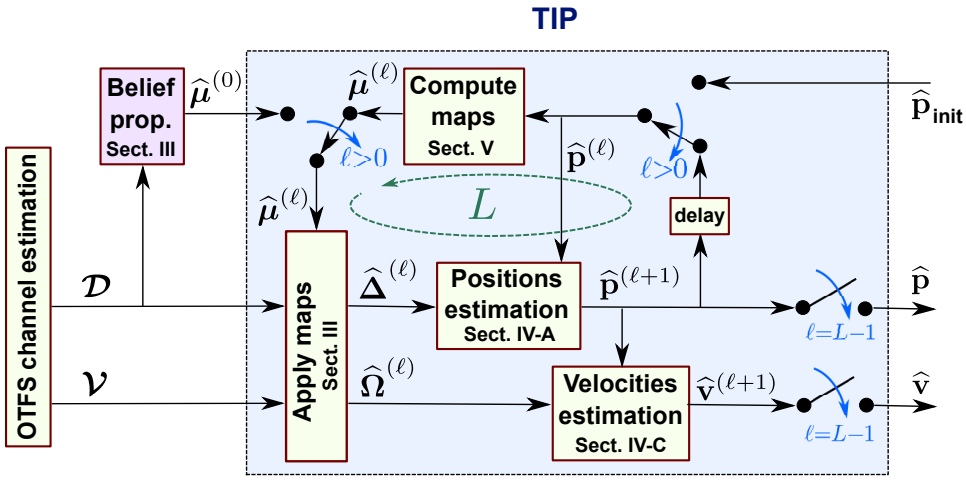

Fig. 2. Scheme of the cold start algorithmic framework. An initial estimate of the maps, $\widehat{\boldsymbol{\mu}}^{(0)}$, is provided by the BP algorithm and fed to TIP. After L iterations, TIP provides an estimate of the UAVs' positions, $\widehat{\mathbf{p}}$, and velocities, $\widehat{\mathbf{v}}$.

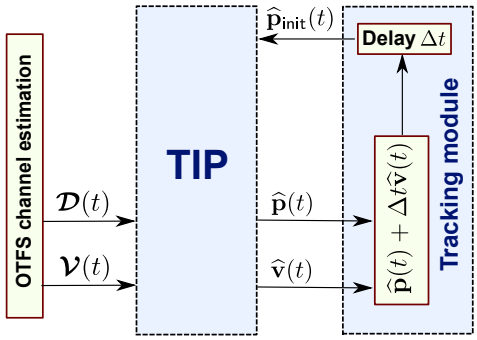

Fig. 3. Scheme of the tracking algorithmic framework. At every time step Δt, TIP is provided with a new set of channel observations $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$. TIP also takes as input an initial estimate of the UAV's positions, obtained at the previous time step and outputs the current estimates $\widehat{\mathbf{p}}(t)$ and $\widehat{\mathbf{v}}(t)$.

the Doppler shifts. As an example, consider signals with carrier frequency f_c=5 GHz and a typical OTFS frame duration T_f=20 ms. Then, the quantization step in the Doppler domain is $1/T_f$=50 Hz. A UAV accelerating at 50 m/s^2 (a rather significant $5g$ acceleration) varies its speed by Δ_v=1 m/s in T_f seconds. This speed variation, assumed to be radial with respect to the observer, would result in a variation in the Doppler shift $\Delta_v f_c/c \approx$16 Hz, which is well below the 50 Hz quantization step and has a negligible impact. Hence, in the following, we assume that estimation errors are only due to quantization noise.

The channel measurement from each UAV can be sent to the edge server using a dedicated control channel. For a swarm composed of N UAVs, there are $N(N-1)$ multipath channels, each of them having $N-1$ paths with their delay and Doppler information. Hence, the total number of measurements is $2N(N-1)^2$, which yields a negligible communication overhead[3].

B. Problem statement and roadmap

If the UAV's positions are unknown, the i-th UAV cannot associate each path in (1) with the identity of the UAV that generated it. Rather, it can only arrange the delays $\widetilde{\delta}_{i,j,k}$ in the list

$$\mathcal{D}_{i,j} = [d_{i,j,1}, \ldots, d_{i,j,N-1}] \tag{8}$$

and the velocities $\widetilde{\omega}_{i,j,k}$ in the list

$$\mathcal{V}_{i,j} = [v_{i,j,1}, \ldots, v_{i,j,N-1}] \tag{9}$$

where the elements of $\mathcal{D}_{i,j}$, as well as of $\mathcal{V}_{i,j}$, appear in increasing order (i.e., $d_{i,j,1} \leq d_{i,j,2} \leq \ldots \leq d_{i,j,N-1}$). In general, $d_{i,j,m}=\widetilde{\delta}_{i,j,k}$, and $v_{i,j,m}=\widetilde{\omega}_{i,j,k}$, if the elements at position m in both lists correspond to the DD signature of the k-th UAV. In other words, we can associate with each list $\mathcal{D}_{i,j}$ and $\mathcal{V}_{i,j}$ a bijective map of the indices

$$\mu_{i,j} : \{1, \ldots, N\} \backslash \{i\} \rightarrow \{1, \ldots, N-1\}$$

such that $\mu_{i,j}(k)=m$. Note that $d_{i,j,1}=0$ corresponds to the LoS path between UAV j and UAV i, since it is the shortest possible relative delay. By defining the set of all lists $\boldsymbol{\mathcal{D}} \triangleq \{\mathcal{D}_{i,j} | i,j=1,\ldots,N,\, i \neq j\}$ and $\boldsymbol{\mathcal{V}} \triangleq \{\mathcal{V}_{i,j} | i,j=1,\ldots,N,\, i \neq j\}$, and the set of all maps $\boldsymbol{\mu} \triangleq \{\mu_{i,j} | i,j=1,\ldots,N,\, i \neq j\}$, the problem addressed in this work can be stated as follows:

Problem statement – *Given N UAVs communicating between each-other using OTFS through channels as in* (1)*, and the lists of noisy delay-Doppler profiles $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$, how can an edge server reliably estimate the UAVs' geometric positions and velocities vectors, $\vec{P}_i$ and $\vec{V}_i$, i=1, …, N?*

[3]For example, assuming each delay and Doppler measurement takes 2 bytes, the total number of bytes required for transferring all lists is $4N(N-1)^2$, which, for a swarm composed of N=20 UAVs adds up to about 30 kbytes. If the updated lists are transferred every second to the edge server, the total required overhead is about 240 kbit/s which can be easily supported by the control channels of modern communication systems.

Solution roadmap. We answer this question by developing a new positioning algorithm called TIP (Turbo Iterative Positioning). Using the RED and Doppler shift measurements of the OTFS channel estimation on all the UAVs links, collected in the set of lists $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$ (resp.), TIP enables two operational modes: (i) *cold start* (Fig. 2) where no prior information on positions and velocities is available, and (ii) *tracking* (Fig. 3) where the estimated positions and velocities are updated every Δt seconds using the previous estimates. The cold start mode is based on a BP algorithm to produce an initial estimate of the maps, $\widehat{\boldsymbol{\mu}}^{(0)}$, assigning the identity of the scatterers in each path of each link.

The TIP algorithm takes as input such initial maps' estimate and iterates L times to refine the estimated positions and velocities, and correct residual errors in the corresponding maps. At the core of the iterative loop of TIP, a gradient descent algorithm minimizes an error function to estimate the positions at step $\ell+1$, given the currently estimated positions and maps. The same maps are applied to the lists of Doppler shifts $\boldsymbol{\mathcal{V}}$ to estimate the velocities. In the tracking mode, the estimates of positions and velocities are combined to predict the next positions after Δt seconds, which can be fed back to TIP as a reliable starting point for the gradient descent algorithm.

The overall approach is detailed in Sec. V–VII and includes the following blocks:

1) *Belief Propagation* – provides an initial estimate of the maps $\boldsymbol{\mu}$, given the sets of lists $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$. Towards this goal, we relax the set of deterministic maps $\boldsymbol{\mu}$ to a set of random maps and use a BP approach to obtain an estimation $\widehat{\boldsymbol{\mu}}^{(0)}$ (see Sec. V).
2) *Apply maps* – permutes the RED and Doppler shift lists $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$ according to the estimated maps $\widehat{\boldsymbol{\mu}}^{(\ell)}$, to produce the permuted lists $\widehat{\boldsymbol{\Delta}}^{(\ell)}$ and $\widehat{\boldsymbol{\Omega}}^{(\ell)}$, respectively (see Sec. V-E).
3) *Positions estimation* – computes the new UAVs' position estimates $\widehat{\mathbf{p}}^{(\ell+1)}$, given the lists $\widehat{\boldsymbol{\Delta}}^{(\ell)}$ from *Apply maps* and $\widehat{\mathbf{p}}^{(\ell)}$, using a gradient descent algorithm (see Sec. VI-A).
4) *Velocity estimation* – provides estimates $\widehat{\mathbf{v}}^{(\ell+1)}$ of the UAV's velocities using the estimates $\widehat{\mathbf{p}}^{(\ell+1)}$, and the set of lists $\widehat{\boldsymbol{\Omega}}^{(\ell)}$ from *Apply maps* (see Sec. VI-C).
5) *Compute maps* – computes the set of maps $\widehat{\boldsymbol{\mu}}^{(\ell)}$ corresponding to the current position estimates $\widehat{\mathbf{p}}^{(\ell)}$ (see Sec. VII).

V. Solving the network assignment problem through BP

To associate channel observations with UAVs' identities in each link, the edge server has to estimate the maps $\boldsymbol{\mu}$. This is an instance of an assignment problem that can be solved by using a probabilistic approach. Due to the presence of noise we can relax the set of deterministic maps $\boldsymbol{\mu}$ to a set of random maps $\boldsymbol{\mathcal{M}} \triangleq \{\mathcal{M}_{i,j} | i,j=1,\ldots,N, i \neq j\}$ having joint posterior distribution $\mathbb{P}(\boldsymbol{\mathcal{M}} | \boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{V}})$, given the channel measurements $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$. The optimal estimation of the maps would require to solve

$$\widehat{\boldsymbol{\mu}} = \arg\max_{\boldsymbol{\mu}} \mathbb{P}(\boldsymbol{\mathcal{M}} = \boldsymbol{\mu} | \boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{V}}) \tag{10}$$

which has an intractable complexity of $O([(N-1)!]^{N(N-1)})$. To overcome this complexity, we propose a BP approach based on a factor graph with specific variable nodes and check nodes [25]. In the rest of this section, we describe how to specialize this proposed approach to our setting. To simplify the presentation, we first restrict BP to the case where only the channel measurements $\boldsymbol{\mathcal{D}}$ are used. Then we extend this to the case where both $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$ are employed in (10).

A. Using check node relations on delays

Let us first focus on a subset of four UAVs, respectively labelled by i,j,k,h, as depicted in Fig. 1. The figure also highlights the actual distances $|\mathbf{p}_u - \mathbf{p}_v|$ between the UAVs, for $u,v \in \{i,j,k,h\}$. By recalling the expression of $\delta_{i,j,k}$ in (3), it is easy to observe that for all quadruples of UAVs in the swarm, we have:

$$\delta_{i,j,k} - \delta_{i,j,h} + \delta_{i,k,h} - \delta_{j,h,k} = 0 \tag{11}$$

where $i,j,k,h \in \{1,\ldots,N\}$, $j \neq i$, $k \neq \{i,j\}$, $h \neq \{i,j,k\}$. This implies that one can find the correct associations $\mu_{i,j}$ between the entries of the list $\mathcal{D}_{i,j}$ and the UAVs' identities by searching through the elements of lists $\mathcal{D}_{i,j}$, $\mathcal{D}_{i,k}$, and $\mathcal{D}_{j,h}$, until a relation such as the one in (11) holds. More precisely, given the ordered lists $\mathcal{D}_{i,j}$, $\mathcal{D}_{i,k}$, and $\mathcal{D}_{j,h}$, if for some integers $m,n,s,t \in \{1,\ldots,N-1\}$ and such that $m \neq n$, the following relation holds in the absence of noise:

$$d_{i,j,m} - d_{i,j,n} + d_{i,k,s} - d_{j,h,t} = 0\,, \tag{12}$$

then, by comparing (11) to (12), we can deduce the maps $\mu_{i,j}(k)=m$, $\mu_{i,j}(h)=n$, $\mu_{i,k}(h)=s$, and $\mu_{j,h}(k)=t$.

It is worth noting that there are $O(N^8)$ constraint equations like (12) that need to be satisfied simultaneously by the d's in all the lists $\boldsymbol{\mathcal{D}}$. Additionally, whenever the distances $\delta_{i,j,k}$ are affected by noise, the left-hand side of (12) is, in general, not zero but random.

B. Defining the variable nodes and computing their marginals

We observe that the random map $\mathcal{M}_{i,j}$ can be represented by a set of $N-1$ correlated discrete random variables $\mathcal{M}_{i,j,k}$, $k=\{1,\ldots,N\}/\{i\}$ having support on the integer set $\{1,\ldots,N-1\}$ and probability mass function (pmf) $\pi_{i,j,k,m} \triangleq \mathbb{P}(\mathcal{M}_{i,j,k}=m|\boldsymbol{\mathcal{D}})$ with $\sum_{m=1}^{N-1} \pi_{i,j,k,m}=1$. These correspond to the variable nodes in the factor graph. The terms $\pi_{i,j,k,m}$ are the marginals of the joint posterior distribution $\mathbb{P}(\boldsymbol{\mathcal{M}} | \boldsymbol{\mathcal{D}})$ and represent the probability that the RED profile of the k-th UAV is identified by the m-th element of the list $\mathcal{D}_{i,j}$. In the absence of quantization noise, $\eta_{i,j,k}$, the pmf of $\mathcal{M}_{i,j,k}$ coincides with a pmf with probability 1 in correspondence with the actual value $\mu_{i,j}(k)$, i.e., the correct identity of the UAV. We first show how to compute such marginals using a BP approach, and then we describe how to estimate the maps $\boldsymbol{\mu}$, given the marginals $\pi_{i,j,k,m}$ in Sec. V-C.

To compute the marginals $\pi_{i,j,k,m}$, we first need to consider some special cases of (12), which must be treated separately.

Specifically, we observe that, for every $i\neq j$, when $m=1$, the first term of (12) is $d_{i,j,1}=0$ by definition since it is the smallest possible RED and, thus, it refers to the LoS path[4], i.e., the path connecting node j with node i. It follows that $\mu_{i,j}(j)=1$, which also implies $k=j$. In this case, (11) becomes meaningless since its left-hand side is identically zero. Similarly, referring to (12), we deduce $\mu_{i,k}(k)=1$ when $s=1$ and $\mu_{j,h}(h)=1$, and $t=1$. Summarizing, for all $i\neq j$, we can handle these special cases by setting $\pi_{i,j,k,1}=1$ for $k=j$ and $\pi_{i,j,k,1}=0$ for $k\neq j$.

Specializing the BP approach to the notation in [25], we only consider the indices m,n,s,t, with $m\neq n$ ranging in $\{2,\ldots,N-1\}$ and factorize the joint posterior pmf $\mathbb{P}(\boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{D}})$ as

$$\mathbb{P}(\boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{D}}) = \prod_{Q\in\mathcal{Q}} \psi_Q\left(\boldsymbol{\mathcal{M}}_Q|\boldsymbol{\mathcal{D}}_Q\right) \tag{13}$$

where

$$\mathcal{Q} = \{[i,j,k,h]|[i,j,k,h]\in\{1,\ldots,N\}^4, [i,j,k,h]_{\neq}\} \tag{14}$$

is the set of all possible quadruples Q of distinct UAVs. For a given $Q=[i,j,k,h]$, $\boldsymbol{\mathcal{M}}_Q\subseteq\boldsymbol{\mathcal{M}}$ denotes the set of random variables $\boldsymbol{\mathcal{M}}_Q=\{\mathcal{M}_{i,j,k},\mathcal{M}_{i,j,h},\mathcal{M}_{i,k,h},\mathcal{M}_{j,h,k}\}$, and $\boldsymbol{\mathcal{D}}_Q=\{\mathcal{D}_{i,j},\mathcal{D}_{i,k},\mathcal{D}_{j,h}\}$. In analogy with (12), the factor $\psi_Q(\boldsymbol{\mathcal{M}}_Q|\boldsymbol{\mathcal{D}}_Q)$ is associated with the equation

$$d_{i,j,\mathcal{M}_{i,j,k}} - d_{i,j,\mathcal{M}_{i,j,h}} + d_{i,k,\mathcal{M}_{i,k,h}} - d_{j,h,\mathcal{M}_{j,h,k}} = 0\,, \tag{15}$$

which depends on the subset of random variables $\boldsymbol{\mathcal{M}}_Q$ and is parameterized by $\boldsymbol{\mathcal{D}}_Q$. In practice, when the random variables in $\boldsymbol{\mathcal{M}}_Q$ take on the specific values $\mathcal{M}_{i,j,k}=m$, $\mathcal{M}_{i,j,h}=n$, $\mathcal{M}_{i,k,h}=s$, $\mathcal{M}_{i,j,k}=t$, (15) converts to (12) and the function $\psi_Q\left(\mathcal{M}_{i,j,k}=m,\mathcal{M}_{i,j,h}=n,\mathcal{M}_{i,k,h}=s,\mathcal{M}_{i,j,k}=t|\boldsymbol{\mathcal{D}}_Q\right)$ provides a measure of the likelihood that equation (12) is satisfied. Note that the left-hand side of (12) contains the sum of 4 independent quantization errors characterized by the uniform distribution $f_\eta(\cdot)$ with support in $[-\frac{c}{2B},\frac{c}{2B}]$. Therefore, the distribution of such sum is $g(\cdot)=f_\eta(\cdot)*f_\eta(\cdot)*f_\eta(\cdot)*f_\eta(\cdot)$ where $*$ denotes the convolution operator. The above definition of $g(\cdot)$ can be explained by the fact that (12) is a sum of four quantized values of actual REDs, which are unknown, but within the quantization intervals. Due to the rounding operation, the quantization error is deterministically dependent upon the actual value of a RED, but the uncertainty of such a value can be translated onto the uniform distribution $f_\eta(\cdot)$ centered around the center point of a quantization interval for the η's to represent a random quantization noise. Summarizing, for any $Q=[i,j,k,h]$ and integers $m,n,s,t\in\{2,\ldots,N-1\}$, $m\neq n$, let $z_Q^{m,n,s,t}$ be the l.h.s. of (12). Then, we can write

$$\psi_Q\left(\boldsymbol{\mathcal{M}}_Q=[m,n,s,t]|\boldsymbol{\mathcal{D}}_Q\right) = g\left(z_Q^{m,n,s,t}\right)\,.$$

[4]Because of quantization noise, some entries in the list $\mathcal{D}_{i,j}$ might take the same value. This means that, by observing the list, the UAVs generating such entries are indistinguishable. We also know that one of the elements of the list is due to the LoS path associated with zero delay. If two or more entries of $\mathcal{D}_{i,j}$ take on the same value due to quantization, we arbitrarily associate one of them with the LoS path. We then rely on the TIP iterations to correct any error in the choice between the two.

For convenience, let us define the set of all possible triples of distinct UAVs

$$\mathcal{T} = \{[i,j,k]|[i,j,k]\in\{1,\ldots,N\}^3, [i,j,k]_{\neq}\}$$

so that each triple $T\in\mathcal{T}$ uniquely identifies the random variable $\mathcal{M}_T$. Then, the factorization in (13) can be represented by a *factor graph* whose *variable nodes* are the random variables $\mathcal{M}_T$, $T\in\mathcal{T}$ and the *factor nodes* or *check nodes* are the functions $\psi_Q(\cdot)$, for all $Q\in\mathcal{Q}$. In such graph each check node $Q\in\mathcal{Q}$ is connected to 4 variable nodes. Specifically, the check node $Q=[i,j,k,h]$ has neighbours $T_1=[i,j,k]$, $T_2=[i,j,h]$, $T_3=[i,k,h]$, and $T_4=[j,h,k]$. Each variable node $T=[i,j,k]\in\mathcal{T}$ has $4(N-3)$ neighbouring check nodes, i.e., $[i,j,k,\ell]$, $[i,j,\ell,k]$, $[i,\ell,j,k]$, and $[\ell,i,j,k]$, for $\ell\in\{1,\ldots,N\}\backslash\{i,j,k\}$. The graph has cycles of a minimum length of 4, since both check nodes $Q_1=[i,j,k,h]$ and $Q_2=[i,j,h,k]$ are connected to the variable nodes $T_1=[i,j,k]$ and $T_2=[i,j,h]$. Therefore, convergence of the BP algorithm cannot be theoretically guaranteed.

Let $\mathcal{N}_2$ be the set $\{2,\ldots,N-1\}$. Then, in the BP algorithm, the messages flowing from the check node Q to the adjacent variable nodes T_1, T_2, T_3, and T_4 are given by

$$\begin{aligned}
\zeta_{Q\to T_1}(m) &= \sum_{\substack{n,s,t\in\mathcal{N}_2\\ n\neq m}} g(z_Q^{m,n,s,t})\lambda_{T_2\to Q}(n)\lambda_{T_3\to Q}(s)\lambda_{T_4\to Q}(t)\\
\zeta_{Q\to T_2}(n) &= \sum_{\substack{m,s,t\in\mathcal{N}_2\\ m\neq n}} g(z_Q^{m,n,s,t})\lambda_{T_1\to Q}(m)\lambda_{T_3\to Q}(s)\lambda_{T_4\to Q}(t)\\
\zeta_{Q\to T_3}(s) &= \sum_{\substack{n,m,t\in\mathcal{N}_2\\ m\neq n}} g(z_Q^{m,n,s,t})\lambda_{T_1\to Q}(m)\lambda_{T_2\to Q}(n)\lambda_{T_4\to Q}(t)\\
\zeta_{Q\to T_4}(t) &= \sum_{\substack{m,n,s\in\mathcal{N}_2\\ m\neq n}} g(z_Q^{m,n,s,t})\lambda_{T_1\to Q}(m)\lambda_{T_2\to Q}(n)\lambda_{T_3\to Q}(s)\,.
\end{aligned} \tag{16}$$

Let $\mathcal{Q}_T\subseteq\mathcal{Q}$ be the set of check nodes connected to the variable node T. The message flowing from the variable node T to check node Q is given by

$$\lambda_{T\to Q}(m) = \prod_{Q'\in\mathcal{Q}_T\backslash\{Q\}} \zeta_{Q'\to T}(m) \tag{17}$$

for all $m\in\mathcal{N}_2$. The procedure is iterative and is initialized by setting $\lambda_{T\to Q}(m)=1/(N-2)$, for $m\in\mathcal{N}_2$. Note that the expressions in (16) and (17) need to be normalized to add up to 1 to form a probability vector with $N-2$ entries. At each iteration, the marginal probabilities at variable node $T=[i,j,k]$, i.e., the beliefs, are computed as

$$\pi_{i,j,k,m} = \frac{\prod_{Q\in\mathcal{Q}_T}\zeta_{Q\to T}(m)}{\sum_{m=2}^{N-1}\prod_{Q\in\mathcal{Q}_T}\zeta_{Q\to T}(m)}\,. \tag{18}$$

The above procedure is summarized in Algorithm 1, which has complexity $O(N^8)$.

C. Estimating the maps $\boldsymbol{\mu}$ given the marginals $\pi_{i,j,k,m}$

To estimate the maps, we propose the heuristic greedy approach in Algorithm 2, which takes as input the marginals $\pi_{i,j,k,m}$ obtained from Algorithm 1 and has complexity $O\left(N(N-1)(N-2)\right)$.

For each pair of UAVs, (i,j), the algorithm collects the marginals $\pi_{i,j,k,m}$ in the $N\times(N-1)$ matrix $\mathbf{\Pi}$. It then works iteratively and, at each step, finds the most likely association in the map. Specifically, it seeks the largest entry of $\mathbf{\Pi}$, records its row index k' and its column index m', and sets $\widehat{\mu}_{i,j}(k')=m'$. Next, the algorithm sets the k'-th row and the m'-th column of $\mathbf{\Pi}$ to zero: this operation is necessary because no other element of the map can be assigned to value m' in the following steps of the algorithm. The procedure ends when all values of the elements of $\mathbf{\Pi}$ become zero, i.e., a decision is made on all the elements of the map $\widehat{\mu}_{i,j}$.

D. Extension using check nodes on Doppler shifts

We now show how the measurements of Doppler shifts can be added to the BP algorithm. First, an additional set of check-nodes equations are added to those described by (11). These new equations involve the Doppler shifts measurements, $\boldsymbol{\mathcal{V}}$, and can be derived by recalling the expression of $\omega_{i,j,k}$ in (4). We have

$$\omega_{i,j,k}+\omega_{i,j,h}-\omega_{k,h,i}-\omega_{k,h,j}=0 \tag{19}$$

for every quadruple $Q\in\mathcal{Q}$ defined in (14). Using both (11) and (19), the joint posterior probability mass function $\mathbb{P}(\boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{V}})$ can be factorized as

$$\mathbb{P}(\boldsymbol{\mathcal{M}}|\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{V}})=\prod_{Q\in\mathcal{Q}}\psi_Q\left(\boldsymbol{\mathcal{M}}_Q|\boldsymbol{\mathcal{D}}_Q\right)\phi_Q\left(\boldsymbol{\mathcal{M}}'_Q|\boldsymbol{\mathcal{V}}_Q\right) \tag{20}$$

where $\boldsymbol{\mathcal{M}}'_Q\subseteq\boldsymbol{\mathcal{M}}$ denotes the set of random variables $\boldsymbol{\mathcal{M}}'_Q=\{\mathcal{M}_{i,j,k},\mathcal{M}_{i,j,h},\mathcal{M}_{k,h,i},\mathcal{M}_{k,h,j}\}$, and $\boldsymbol{\mathcal{V}}_Q=\{\mathcal{V}_{i,j},\mathcal{V}_{k,h}\}$. Similarly to the discussion following (15), we define

$$\phi_Q\left(\boldsymbol{\mathcal{M}}'_Q=[m,n,s,t]|\boldsymbol{\mathcal{V}}_Q\right)=g'\left(w_Q^{m,n,s,t}\right)$$

where $g'(\cdot)=f_\zeta(\cdot)*f_\zeta(\cdot)*f_\zeta(\cdot)*f_\zeta(\cdot)$ and $f_\zeta(\cdot)$ is a uniform distribution with support $[-\frac{c}{2T_f f_c},\frac{c}{2T_f f_c}]$. Furthermore, $w_Q^{m,n,s,t}$ is given by

$$w_Q^{m,n,s,t}=v_{i,j,m}+v_{i,j,n}-v_{k,h,s}-v_{k,h,t}$$

for the integers $m,n,s,t\in\{1,\dots,N-1\}$ such that $m\neq n$ and $s\neq t$. Without going into the details, (16)–(18) can then be modified to account for the additional check nodes. Algorithms 1 and 2 need to be modified accordingly.

E. Applying the estimated maps to lists $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$

Once the estimated map $\widehat{\mu}_{i,j}$ is available, it can be applied to lists $\mathcal{D}_{i,j}$ and $\mathcal{V}_{i,j}$ in order to associate to each reflecting UAV, k, an estimate of the distance $\delta_{i,j,k}$ and of the velocity $\omega_{i,j,k}$. We recall that the lists $\mathcal{D}_{i,j}$ and $\mathcal{V}_{i,j}$ are defined in (8) and (9), respectively.

Specifically, the distance associated with the k-th path is estimated as $\widehat{\delta}_{i,j,k}=d_{i,j,\widehat{m}}$ where $\widehat{m}=\widehat{\mu}_{i,j}(k)$, and $d_{i,j,\widehat{m}}$ is an element of the list $\mathcal{D}_{i,j}$. Similarly, the velocity associated with the k-th path is estimated as $\widehat{\omega}_{i,j,k}=v_{i,j,\widehat{m}}$ where $v_{i,j,\widehat{m}}$ is an element of the list $\mathcal{V}_{i,j}$. This procedure is represented by the block labeled "Apply maps", shown in Fig. 2.

Algorithm 1 $\{\pi_{i,j,k,m}\}=$ ComputeMarginals($\boldsymbol{\mathcal{D}}$)

Require: N, $\boldsymbol{\mathcal{D}}$, $I_\mu>0$
  for $i,j=1,\dots,N$, $j\neq i$ **do**
    $\pi_{i,j,k,1}\leftarrow 1$, for $k=j$ and $\pi_{i,j,k,1}\leftarrow 0$ for $k\neq\{i,j\}$
  for $T\in\mathcal{T}$ **do**
    for $Q\in\mathcal{Q}$ **do**
      for $m=2,\dots,N-1$ **do**
        initialize $\lambda_{T\to Q}(m)=\frac{1}{N-2}$
  for $\ell=1,\dots,I_\mu$ **do**
    for $Q\in\mathcal{Q}$ **do**
      let $\mathcal{T}_Q$ be the neighbours of Q
      for $T\in\mathcal{T}_Q$ **do**
        for $b=2,\dots N-1$ **do**
          compute $\zeta_{Q\to T}(b)$ according to (16)
    for $T\in Tc$ **do**
      let $\mathcal{Q}_T$ be the neighbours of T
      for $Q\in\mathcal{Q}_T$ **do**
        for $b=2,\dots N-1$ **do**
          compute $\lambda_{T\to Q}(b)$ according to (17)
  for $T\in\mathcal{T}$ **do**
    let $\mathcal{Q}_T$ be the neighbours of T
    let $T=[i,j,k]$
    for $m=2,\dots,N-1$ **do**
      compute the marginals $\pi_{i,j,k,m}$ using (18)
  return $\{\pi_{i,j,m,k}\}$

Algorithm 2 $\widehat{\boldsymbol{\mu}}=$ EstimateMaps($\boldsymbol{\mathcal{D}}$)

Require: N,$\boldsymbol{\mathcal{D}}$
  $\{\pi_{i,j,k,m}\}\leftarrow$ ComputeMarginals($\boldsymbol{\mathcal{D}}$)
  for $i,j=1,\dots,N$, $i\neq j$ **do**
    $[\mathbf{\Pi}]_{k,m}\leftarrow\begin{cases}\pi_{i,j,k,m} & k\neq i\\ 0 & k=i\end{cases}$
    while $\mathbf{\Pi}\neq\mathbf{0}$ **do**
      $[k',m']\leftarrow\arg\max_{k,m}([\mathbf{\Pi}]_{k,m})$
      $\widehat{\mu}_{i,j}(k')\leftarrow m'$
      $[\mathbf{\Pi}]_{q,m'}\leftarrow 0$ for $q=1,\dots,N$
      $[\mathbf{\Pi}]_{k',q}\leftarrow 0$ for $q=1,\dots,N-1$
  return $\widehat{\boldsymbol{\mu}}$

VI. Estimating the Node Positions and Velocities

Once the estimates $\widehat{\boldsymbol{\mu}}$ of the maps $\boldsymbol{\mu}$ have been used by the edge server to associate the elements of the lists $\boldsymbol{\mathcal{D}}$ with the UAVs that generated them (see Sec V-E), we apply the gradient descent (GD) algorithm to minimize the square error, $\mathcal{E}$, between $\widehat{\delta}_{i,j,k}$ and the tentative channel observations $\theta_{i,j,k}$. We remark that, in the case of perfect maps estimates, the edge server can retrieve the channel observations $\widehat{\delta}_{i,j,k}=\widetilde{\delta}_{i,j,k}$ in their correct ordering.

A. Gradient descent algorithm

Without loss of generality, we describe the gradient descent algorithm under the assumption that the maps have been correctly estimated. Let $\mathbf{t}_i=[t_{i,x}, t_{i,y}, t_{i,z}]^\mathsf{T}$ be a tentative decision for the position of UAV i, $\mathbf{p}_i$, and let $\mathbf{t}\triangleq[\mathbf{t}_1^\mathsf{T}, \ldots, \mathbf{t}_N^\mathsf{T}]^\mathsf{T}$. Then, an estimate $\widehat{\mathbf{p}}$ of the actual components $\mathbf{p}$ can be obtained by solving the problem

$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{t}} \mathcal{E}(\mathbf{t}) \tag{21}$$

where

$$\mathcal{E}(\mathbf{t}) \triangleq \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{\substack{k=1 \\ k\neq i,j}}^{N} \left(\widetilde{\delta}_{i,j,k} - \theta_{i,j,k}(\mathbf{t})\right)^2 \tag{22}$$

is the square error between the channel observations, $\widetilde{\delta}_{i,j,k}$, and the tentative distances $\theta_{i,j,k}(\mathbf{t})\triangleq|\mathbf{t}_j-\mathbf{t}_k|+|\mathbf{t}_k-\mathbf{t}_i|-|\mathbf{t}_j-\mathbf{t}_i|$ according to (3). The square error in (22) is, in general, a non-convex function, nevertheless local or global minima can be easily found by applying a standard GD method.

In the GD algorithm, let $\mathbf{t}^{(\alpha)}$ be the tentative positions at iteration $\alpha\geq 0$. Then, after applying one GD step, the tentative estimated position at iteration $\alpha+1$ is updated as $\mathbf{t}^{(\alpha+1)}=\mathbf{t}^{(\alpha)}-\gamma^{(\alpha)}\mathbf{g}^{(\alpha)}$ where $\gamma^{(\alpha)}$ denotes the step size, $\mathbf{g}\triangleq\frac{\partial}{\partial\mathbf{t}}\mathcal{E}(\mathbf{t})$ is the gradient of $\mathcal{E}(\mathbf{t})$ and $\mathbf{g}^{(\alpha)}=\mathbf{g}|_{\mathbf{t}=\mathbf{t}^{(\alpha)}}$. The step size can be kept constant or selected at each iteration according to some rule as, e.g., the Barzilai-Borwein method, where $\gamma^{(\alpha)}$ is computed by exploiting the trend of the most recent two iterations [30]. Let $\mathbf{g}\triangleq[\mathbf{g}_1^\mathsf{T}, \ldots, \mathbf{g}_N^\mathsf{T}]^\mathsf{T}$. Then $\mathbf{g}_h$, $h=1,\ldots,N$, is given by

$$\mathbf{g}_h(\mathbf{t}) = \frac{\partial \mathcal{E}(\mathbf{t})}{\partial \mathbf{t}_h} = -2 \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{\substack{k=1 \\ k\neq i,j}}^{N} w_{i,j,k}(\mathbf{t}) \frac{\partial \theta_{i,j,k}(\mathbf{t})}{\partial \mathbf{t}_h} \tag{23}$$

where $w_{i,j,k}(\mathbf{t})\triangleq\widetilde{\delta}_{i,j,k}-\theta_{i,j,k}(\mathbf{t})$. Since $\frac{\partial|\mathbf{x}|}{\partial\mathbf{x}}=\frac{\mathbf{x}}{|\mathbf{x}|}$, for any vector $\mathbf{x}$, using (3) we have

$$\frac{\partial \theta_{i,j,k}(\mathbf{t})}{\partial \mathbf{t}_h} = \begin{cases} \mathbf{u}_{i,k}(\mathbf{t}) - \mathbf{u}_{i,j}(\mathbf{t}) & \text{if } i=h \\ \mathbf{u}_{j,k}(\mathbf{t}) - \mathbf{u}_{j,i}(\mathbf{t}) & \text{if } j=h \\ \mathbf{u}_{k,i}(\mathbf{t}) + \mathbf{u}_{k,j}(\mathbf{t}) & \text{if } k=h \\ \mathbf{0} & \text{else} \end{cases} \tag{24}$$

where the versors $\mathbf{u}_{n,m}$ are defined in (5). By using (24) in (23), we obtain

$$\mathbf{g}_h = -2 \begin{cases} \sum_{j\neq h} \sum_{k\neq h,j} w_{h,j,k}(\mathbf{u}_{h,k} - \mathbf{u}_{h,j}) & \text{if } i=h \\ \sum_{i\neq h} \sum_{k\neq i,h} w_{i,h,k}(\mathbf{u}_{h,k} - \mathbf{u}_{h,i}) & \text{if } j=h \\ \sum_{i\neq h} \sum_{j\neq i,h} w_{i,j,h}(\mathbf{u}_{h,i} + \mathbf{u}_{h,j}) & \text{if } k=h\,. \end{cases}$$

By summing the above three contributions and renaming the summation indices, after some algebra, we obtain

$$\mathbf{g}_h = 2 \sum_{\substack{i=1 \\ i\neq h}}^{N} \sum_{\substack{j=1 \\ j\neq h,i}}^{N} (w_{h,i,j}+w_{i,h,j})(\mathbf{u}_{h,i}-\mathbf{u}_{h,j}) - w_{i,j,h}(\mathbf{u}_{h,i}+\mathbf{u}_{h,j})$$

where, for simplicity, we omitted the dependency on $\mathbf{t}$. In our implementation, the GD algorithm starts from an initial tentative value, $\mathbf{t}^{(0)}$, and iterates until the following stopping condition on the square error is satisfied:

$$\frac{\mathcal{E}(\mathbf{t}^{(\alpha+1)}) - \mathcal{E}(\mathbf{t}^{(\alpha)})}{\mathcal{E}(\mathbf{t}^{(\alpha)})} < \epsilon\,,$$

or a maximum number of iterations, I_α, is reached. The initial tentative $\mathbf{t}^{(0)}$ can be (i) chosen either as a prior estimate of the position or (ii) randomly drawn. Case (i) is selected when TIP operates in tracking mode, or in cold star from the second TIP iteration onwards; case (ii) is selected only in cold start mode at the first TIP iteration.

B. Improving convergence to global optimum

Since the square error function $\mathcal{E}(\mathbf{t})$ is not convex, the GD algorithm may reach a local minimum instead of the global one, thus leading to inaccurate positioning. To ensure convergence to the global minimum with high probability, we propose a heuristic method based on the following considerations. Consider that the GD algorithm has converged to its global minimum and that the minimizer of (22) is $\mathbf{t}=\widehat{\mathbf{p}}$. Such estimate of the actual positions $\mathbf{p}$ can be written as $\widehat{\mathbf{p}}=\mathbf{p}+\mathbf{e}$ where $\mathbf{e}$ is the estimation error. Hence,

$$\theta_{i,j,k}(\widehat{\mathbf{p}}) = \theta_{i,j,k}(\mathbf{p}) + \psi_{i,j,k}$$

where, by definition, $\theta_{i,j,k}(\mathbf{p})=\delta_{i,j,k}$ and $\psi_{i,j,k}$ is the estimation error in the RED. Using (6), the term in the sum in (22) reduces to $\widetilde{\delta}_{i,j,k}-\theta_{i,j,k}(\widehat{\mathbf{p}})=\eta_{i,j,k}-\psi_{i,j,k}$ where we recall that the terms $\eta_{i,j,k}$ are the quantization errors on the channel observations. Thus, the square error at convergence is

$$\mathcal{E}(\widehat{\mathbf{p}}) = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{\substack{k=1 \\ k\neq i,j}}^{N} \left(\eta_{i,j,k} - \psi_{i,j,k}\right)^2\,. \tag{25}$$

Since we assume that the global minimum of the function $\mathcal{E}(\mathbf{t})$ has been reached at $\mathbf{t}=\widehat{\mathbf{p}}$, any other choice for $\mathbf{t}$ will lead to a larger square error. Consider now the choice $\mathbf{t}=\mathbf{p}$, which corresponds to $\psi_{i,j,k}=0$, $\forall i,j,k$. According to (25), we then have

$$\mathcal{E}(\widehat{\mathbf{p}}) \leq \mathcal{E}(\mathbf{p}) = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{\substack{k=1 \\ k\neq i,j}}^{N} \eta_{i,j,k}^2\,. \tag{26}$$

Now, given the actual position components, $\mathbf{p}$, the quantization errors, $\eta_{i,j,k}$, are deterministic quantities; however, as already observed, the uncertainty in $\mathbf{p}$ can be translated into $\eta_{i,j,k}$ being independent random variables with zero-mean, uniform distribution in the interval $[-\frac{c}{2B}, \frac{c}{2B}]$ and, hence, variance $\sigma_\eta^2=\frac{c^2}{12B^2}$. For sufficiently large N (that is, a sufficiently large number of observations $\widetilde{\delta}_{i,j,k}$), the following holds with high probability:

$$\mathcal{E}(\widehat{\mathbf{p}}) \leq N(N-1)(N-2)\frac{c^2}{12B^2} \triangleq E_b\,. \tag{27}$$

In summary, when the GD algorithm converges to the global minimum, the residual square error is expected to be lower than E_b, whereas we expect $\mathcal{E}(\widehat{\mathbf{p}})$ to be much larger when it converges to a local minimum. Based on the above considerations, we improve the positioning accuracy of our algorithm

by comparing the square error $\mathcal{E}(\widehat{\mathbf{p}})$ at the output of the GD algorithm against the threshold $\widetilde{E}_b{=}\beta E_b$ where $\beta{>}1$ is a constant.

Referring to Fig. 2, the GD verifies the condition $\mathcal{E}(\widehat{\mathbf{p}}){>}\widetilde{E}_b$ at the L-th iteration. If it is not met, the TIP is run again resetting $\ell{=}0$ and using a new randomly drawn $\widehat{\mathbf{p}}^{(0)}_{\text{cold start}}$. This restart is repeated until $\mathcal{E}(\widehat{\mathbf{p}}){\leq}\widetilde{E}_b$, or a maximum number of retries has been reached. Otherwise, failure is declared. Note that the purpose of the constant β is to compensate for possible large statistical deviations of $\mathcal{E}(\mathbf{p})$ (i.e., the r.h.s. of (26)) from E_b. As β decreases, the probability that reliable estimates are declared unreliable increases as well as the computational complexity of the algorithm. Conversely, as β increases, convergence to local minima are less likely to be detected and, in general, the positioning accuracy decreases.

Remark 1: The GD algorithm proposed above allows the joint estimation of the positions of all UAVs. However, we recall that only the positions of $\bar{N}{=}N-A$ UAVs have to be estimated since the remaining A are anchors whose positions are perfectly known. To take this into account, the GD algorithm needs to be slightly modified as follows. Let $\mathcal{A}$ be the set of anchor nodes. Then, for all $i{\in}\mathcal{A}$, the tentative decisions are set to the actual positions, i.e., $\mathbf{t}_i^{(0)}{=}\mathbf{p}_i$ and the gradient $\mathbf{g}_i^{(\alpha)}$ is set to $\mathbf{0}$ at every iteration step α. This ensures $\mathbf{t}_i^{(\alpha)}{=}\mathbf{p}_i$ for all iteration steps. The GD algorithm remains unchanged since the estimation error $\mathbf{e}{=}\widehat{\mathbf{p}}{-}\mathbf{p}$ is zero for all anchor nodes, while it is (in general) non-zero for the others.

C. Estimating the nodes' velocities

Once the estimates of the UAVs' position vectors $\widehat{\mathbf{p}}$ and of the maps $\widehat{\boldsymbol{\mu}}$ are available, the components $\mathbf{v}_i$ of the UAVs' geometric velocities $\vec{V}_i$, $i{=}1,\ldots,N$ can be estimated through (4). Indeed, from (4), we can write

$$\begin{aligned}\omega_{i,j,k} &= \mathbf{u}_{j,k}^{\mathsf{T}}\mathbf{v}_j + (\mathbf{u}_{k,i}-\mathbf{u}_{j,k})^{\mathsf{T}}\mathbf{v}_k - \mathbf{u}_{k,i}^{\mathsf{T}}\mathbf{v}_i \\ &= \mathbf{u}_{i,j,k}^{\mathsf{T}}\mathbf{v} \end{aligned} \tag{28}$$

where $\mathbf{v}{=}[\mathbf{v}_1^{\mathsf{T}},\ldots,\mathbf{v}_N^{\mathsf{T}}]^{\mathsf{T}}$. If we collect the terms $\omega_{i,j,k}$ and $\widetilde{\omega}_{i,j,k}$ in the column vectors $\boldsymbol{\omega}$ and $\widetilde{\boldsymbol{\omega}}$, respectively, according to (7), we can write

$$\widetilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\eta} = \mathbf{U}^{\mathsf{T}}\mathbf{v} + \boldsymbol{\zeta}\,. \tag{29}$$

where the columns of the matrix $\mathbf{U}$ are $\mathbf{u}_{i,j,k}$ and the elements of $\boldsymbol{\zeta}$ are the discretization errors $\zeta_{i,j,k}$. The matrix $\mathbf{U}$ has size $3N{\times}N(N{-}1)^2$, since $N(N{-}1)^2$ is the total number of observations and $3N$ is the total number of components of the velocity vectors.

Since the positions and velocities are known for the A anchors, the edge server only needs to estimate $3\bar{N}$ parameters, with $\bar{N}{=}N{-}A$. In practice, if we let $\bar{\mathcal{N}}$ be the set of the UAVs which are not anchors, we just need to consider the reduced equation

$$\widetilde{\boldsymbol{\omega}} = \bar{\mathbf{U}}^{\mathsf{T}}\bar{\mathbf{v}} + \boldsymbol{\zeta}\,, \tag{30}$$

instead of (29). In (30), $\bar{\mathbf{U}}$ is a $N(N{-}1)^2{\times}3\bar{N}$ matrix containing only the rows of $\mathbf{U}$ corresponding to the UAVs in $\bar{\mathcal{N}}$ and, similarly, $\bar{\mathbf{v}}$ has size $3\bar{N}$ and contains only the contributions of the $\bar{N}$ UAVs in $\bar{\mathcal{N}}$.

We recall that $\boldsymbol{\zeta}$ represents the quantization errors on the observed velocities and that, due to the rounding operation, they are deterministically dependent upon $\boldsymbol{\omega}$ but, as explained in Sec. V, the uncertainty in the value of $\boldsymbol{\omega}$ translates into $\boldsymbol{\zeta}$ behaving as i.i.d. random variables with zero mean and uniform distribution.

If no statistical information about $\bar{\mathbf{v}}$ is available, the least square estimator $\widehat{\bar{\mathbf{v}}} = \left(\bar{\mathbf{U}}\bar{\mathbf{U}}^{\mathsf{T}}\right)^{-1}\bar{\mathbf{U}}\widetilde{\boldsymbol{\omega}}$

could be exploited[5]. However, the above least square estimator cannot directly be applied for the following two reasons:

- The matrix $\bar{\mathbf{U}}$ is a function of the unknown vectors $\mathbf{p}_i$, $i\in\bar{\mathcal{N}}$, for which only estimates are available. Then, in practice, we need to replace $\bar{\mathbf{U}}$ in the above expression with the matrix $\widehat{\bar{\mathbf{U}}}$ computed using the estimated positions $\widehat{\mathbf{p}}_i$, $i\in\bar{\mathcal{N}}$, obtained at the output of TIP.
- The observations $\widetilde{\omega}_{i,j,k}$ are only available as the elements $v_{i,j,m}$ of the ordered lists $\mathcal{V}_{i,j}$. We recall that the relation between the terms $\widetilde{\omega}_{i,j,k}$ and $v_{i,j,m}$ is completely defined by the maps $\boldsymbol{\mu}$, which are unknown. Hence, in a practical implementation, one must replace $\widetilde{\boldsymbol{\omega}}$ with $\widetilde{\boldsymbol{\omega}}^{\star}$, obtained by reordering the lists $\boldsymbol{\mathcal{V}}$ using the estimated maps $\widehat{\boldsymbol{\mu}}$.

It follows that the estimate of $\bar{\mathbf{v}}$ can be obtained as

$$\widehat{\bar{\mathbf{v}}} = \left(\widehat{\bar{\mathbf{U}}}\widehat{\bar{\mathbf{U}}}^{\mathsf{T}}\right)^{-1}\widehat{\bar{\mathbf{U}}}\widetilde{\boldsymbol{\omega}}^{\star}\,, \tag{31}$$

and it is affected by three sources of errors: the quantization error represented by the terms $\zeta_{i,j,k}$, the position error $\mathbf{p}-\widehat{\mathbf{p}}$, and the error in estimating the maps $\boldsymbol{\mu}$.

VII. Turbo Iterative UAV Positioning and Tracking

The TIP algorithm schematized in Fig. 2 combines the methods presented in Section VI, in a iterative solution for estimating both positions and velocities of the UAVs. Here, we provide a detailed description of the TIP operational modes. In particular, first Sec. VII-A introduces the *cold start mode*, a version of TIP designed for the most challenging case where no prior knowledge about positions, velocities, or trajectories of the UAVs is available to the system. Then Sec. VII-B describes the *tracking mode*, where UAVs follow a trajectory and provide channel profiles to the edge server once every Δt seconds. At each time step, TIP exploits the current channel profiles and the previous estimates of the positions/velocities, to infer the current positions and velocities.

A. Cold start mode

The pseudocode for TIP working in cold start mode is outlined in Algorithm 3. TIP takes as input the set of lists $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$, which are collected by the edge server. We recall that such lists contain the discretized delay-Doppler profiles of all channels connecting any pair of UAVs. As explained in Sec. IV, the elements of list $\mathcal{D}_{i,j}$ are the estimated path distances $d_{i,j,m}$, $m{=}1,\ldots,N{-}1$, ordered in increasing order. The first main challenge for estimating the UAV positions is

[5]When instead a prior distribution for $\bar{\mathbf{v}}$ is known, better estimators (e.g., the maximum a posteriori estimator) can be used.

to associate each path distance $d_{i,j,m}$ with the identity of the UAV that generated it, i.e., to find the maps $\boldsymbol{\mu}$. These maps cannot be obtained deterministically, rather they have to be estimated.

An initial estimate, $\boldsymbol{\mu}^{(0)}$, of the maps is provided using the BP approach described in Algorithms 1 and 2. At the core of the TIP algorithm there is a sophisticated mechanism designed for iteratively refining such maps. In the following, we denote by $\widehat{\boldsymbol{\mu}}^{(\ell)}$ the set of estimated maps at iteration ℓ, which are employed to reorder the lists $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$ according to the rule specified in Sec. V-E. Specifically, $\widehat{\mu}_{i,j}^{(\ell)}$ is applied to $\mathcal{D}_{i,j}$ and $\mathcal{V}_{i,j}$ so as to reorder their elements according to the UAVs' identity index. In doing so, we obtain lists $\widehat{\Delta}_{i,j}^{(\ell)}$ and $\widehat{\Omega}_{i,j}^{(\ell)}$, respectively, where their k-th elements, $\widehat{\delta}_{i,j,k}^{(\ell)}$ and $\widehat{\omega}_{i,j,k}^{(\ell)}$, are an estimate of the distance $\delta_{i,j,k}$ and of the velocity $\omega_{i,j,k}$, respectively. If the map is perfectly estimated, i.e., $\widehat{\mu}_{i,j}^{(\ell)}{=}\mu_{i,j}$, then $\widehat{\delta}_{i,j,k}^{(\ell)}{=}\widetilde{\delta}_{i,j,k}$ and $\widehat{\omega}_{i,j,k}^{(\ell)}{=}\widetilde{\omega}_{i,j,k}$.

The set of reordered lists, $\widehat{\boldsymbol{\Delta}}^{(\ell)}{=}\{\widehat{\Delta}_{i,j}^{(\ell)}|i,j{=}1\ldots,N,i\neq j\}$, is then fed to the GD algorithm described in Sec. VI-A. The GD algorithm is initialized with estimate $\widehat{\mathbf{p}}^{(\ell)}$ at iteration $\ell>0$, and with a random vector $\widehat{\mathbf{p}}_{\mathsf{init}}$ at iteration $\ell=0$. The GD algorithm then outputs the estimated positions, $\widehat{\mathbf{p}}^{(\ell+1)}$, which are employed to (i) provide an estimate of the velocities, $\widehat{\mathbf{v}}^{(\ell+1)}$, using the reordered set of lists $\widehat{\boldsymbol{\Omega}}^{(\ell)}{=}\{\widehat{\Omega}_{i,j}^{(\ell)}|i,j{=}1\ldots,N,i{\neq}j\}$ and the procedure described in Sec. VI-C and (ii) update the estimate of the maps.

This latter task, performed by the block labeled "Compute maps" in Fig. 2, follows the procedure outlined below:

1) from the estimated positions $\widehat{\mathbf{p}}^{(\ell)}$ and for each $i{\neq}j$, we compute the distances $\delta_{i,j,k}^{\star}$ using (3), as
$$\delta_{i,j,k}^{\star}{=}|\widehat{\mathbf{p}}_j^{(\ell)}-\widehat{\mathbf{p}}_k^{(\ell)}|{+}|\widehat{\mathbf{p}}_k^{(\ell)}{-}\widehat{\mathbf{p}}_i^{(\ell)}|{-}|\widehat{\mathbf{p}}_j^{(\ell)}{-}\widehat{\mathbf{p}}_i^{(\ell)}|\,, \quad (32)$$
for $k{\in}\{1,\ldots,N\}\backslash\{i\}$, and we arrange them in the list $\Delta_{i,j}^{\star}$;
2) the elements of $\Delta_{i,j}^{\star}$ are then ordered in ascending order to form the list $\widetilde{\Delta}_{i,j}^{\star}$;
3) finally, the new estimated map $\widehat{\mu}_{i,j}^{(\ell)}$ is defined as the rule that transforms $\Delta_{i,j}^{\star}$ into $\widetilde{\Delta}_{i,j}^{\star}$.

The new maps are then employed to provide a better reordering of the lists $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{V}}$. The algorithm proceeds iteratively until a desired number of iterations, L, has been performed.

Algorithm 3 TIP algorithm: Cold start mode

Require: $\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{V}}$, L, $\widehat{\boldsymbol{\mu}}^{(0)}{\leftarrow}\mathrm{EstimateMaps}(\boldsymbol{\mathcal{D}})$, $\widehat{\mathbf{p}}_{\mathsf{init}}{\leftarrow}\mathrm{rand}$
  for $\ell\leftarrow 0$ to $L-1$ **do**
    if $\ell=0$ **then**
      $\widehat{\mathbf{p}}^{(0)}\leftarrow\widehat{\mathbf{p}}_{\mathsf{init}}$
    else
      for $i,j=1,\ldots,N$, $j\neq i$ **do**
        Create the lists $\Delta_{i,j}^{\star}$ using $\widehat{\mathbf{p}}^{(\ell)}$ and (32)
        Sort each list $\Delta_{i,j}^{\star}$ in ascending order to obtain $\widetilde{\Delta}_{i,j}^{\star}$
        Compute the map $\widehat{\mu}_{i,j}^{(\ell)}$ that yield $\widetilde{\Delta}_{i,j}^{\star}$ from $\Delta_{i,j}^{\star}$
    Apply the maps $\widehat{\boldsymbol{\mu}}^{(\ell)}$ to $\boldsymbol{\mathcal{D}}$ to obtain $\widehat{\boldsymbol{\Delta}}^{(\ell)}$
    Apply the maps $\widehat{\boldsymbol{\mu}}^{(\ell)}$ to $\boldsymbol{\mathcal{V}}$ to obtain $\widehat{\boldsymbol{\Omega}}^{(\ell)}$
    $\widehat{\mathbf{p}}^{(\ell+1)}\leftarrow\mathrm{GradientDescent}(\widehat{\boldsymbol{\Delta}}^{(\ell)},\widehat{\mathbf{p}}^{(\ell)})$
    Compute $\widehat{\mathbf{v}}^{(\ell+1)}$ using (31)
    $\ell\leftarrow\ell+1$
  return $\widehat{\mathbf{p}}^{(L)}$

B. Tracking mode

We now consider the case where each UAV follows a trajectory in space that can be described by the position vectors $\vec{P}_i(t)$ and velocity vectors $\vec{V}_i(t){=}\frac{\mathrm{d}}{\mathrm{d}t}\vec{P}_i(t)$, which are functions of time t. While moving along the trajectory, UAVs periodically send to the edge server (say every Δt seconds) the channel profiles obtained through the OTFS channel estimation, which occurs within a time frame of duration. In such a short time we do not expect position and velocity of the UAVs to change significantly.

Then, at time t, TIP takes as input the set of lists $\boldsymbol{\mathcal{D}}(t)$ and $\boldsymbol{\mathcal{V}}(t)$ and provides an estimate of the positions $\widehat{\mathbf{p}}(t)$ and velocities $\widehat{\mathbf{v}}(t)$. To ease this task, the algorithm also exploits an estimate of the UAV positions previously obtained at time $t-\Delta t$.

In this scenario, TIP is connected to the tracking module depicted in Fig. 3. Also, since a prior estimate of the UAV positions, $\mathbf{p}_{\mathsf{init}}(t)$, is already available, the block named "Belief propagation" is not activated. The reason for this choice is that BP is able to infer reliable estimates of the maps when no prior information on the UAVs' positions are available at a price of computational complexity $O(N^8)$. However, in tracking mode, the system has prior estimates of the UAVs' positions, which makes the use of BP unnecessary. Apart from this aspect, the tracking module works as described in Sec. VII-A.

Summarizing, in tracking mode, TIP works as follows. At time t, TIP

- gets as input the channel profiles $\boldsymbol{\mathcal{D}}(t)$ and $\boldsymbol{\mathcal{V}}(t)$, as well as the prior estimate of the positions $\mathbf{p}_{\mathsf{init}}(t)$ obtained at time $t{-}\Delta t$;
- provides a current estimate of the positions $\widehat{\mathbf{p}}(t)$ and of the velocities $\widehat{\mathbf{v}}(t)$;
- the tracking module provides a forecast for the UAV positions at time $t{+}\Delta t$, computed as
$$\mathbf{p}_{\mathsf{init}}(t+\Delta t)=\widehat{\mathbf{p}}(t)+\Delta t\widehat{\mathbf{v}}(t)\,.$$

C. Genie aided TIP

As a benchmark for the performance of our proposed TIP, we consider a genie-aided (GA) version of TIP which has perfect knowledge of the maps $\boldsymbol{\mu}$. The structure of the GA is a simplification of the scheme in Fig. 2 where $L{=}0$ (no TIP iterations are performed) and the block "Apply maps" takes as input the actual maps, i.e., $\widehat{\boldsymbol{\mu}}^{(0)}{=}\boldsymbol{\mu}$. Consequently, the block "Belief propagation" is not required and the TIP block "Compute maps" is unnecessary. As shown in Sec. IX, the GA version of TIP will be used to provide the lower bound on the error for positions and velocities, since error-free maps are used.

D. Overall solution complexity

We now discuss the complexity of our proposed solution. As mentioned in Sec. V, the BP algorithm designed to associate paths with UAV identities has complexity $O(N^8)$. However, this does not prevent its application to large swarms of UAVs. Large UAV swarms can be handled using a frequency division approach. In practice, a swarm composed of N UAVs can be first partitioned into smaller groups of M UAVs each, and, to avoid interference, a different frequency band can be assigned to each group for communication. By doing so, the complexity of the BP algorithm applied to each group is $O(M^8)$. Then, for constant M, the overall complexity of the BP algorithm is just $O(N)$. The assignment of UAVs to groups can later be modified to increase diversity and improve localization performance in the tracking mode.

For the complexity of the TIP algorithm, we refer to Fig. 2.

- The position estimation block implements the gradient descent algorithm, which iteratively updates the energy $\mathcal{E}(\mathbf{p})$ in (22) and the tentative positions $\mathbf{t}$. The complexity of computing $\mathcal{E}(\mathbf{p})$ and all gradients $\mathbf{g}_h$ is $O(N^3)$ as three summations from 1 to N are involved in their expressions. The gradient descent algorithm stops after reaching convergence. This requires a variable number of iterations, I_α, depending on the values $\widetilde{\delta}_{i,j,k}$. However, I_α does not depend on N, and we can limit it to a maximum value.
- The complexity of estimating the velocities is due to the computation of (31), which requires the inversion of a $3\bar{N} \times 3\bar{N}$ matrix and, thus has complexity $O(N^3)$.
- The block "Apply maps" has also complexity $O(N^3)$, since it involves the permutation of $N(N-1)$ lists of $N-1$ elements.
- Finally, the "Compute Maps" block follows the procedure outlined in Sec. VII-A, which evaluates the estimated distances (32) for all $i,j,k \in 1,\ldots,N$ and then sorts the lists $\Delta^\star_{i,j}$. Again, the complexity of this block is $O(N^3)$.

In conclusion, the overall complexity of TIP is $O(N^3)$. We demonstrate the convergence of BP and TIP in the numerical examples in Sec. IX.

VIII. Cramér Rao lower bounds

To provide a benchmark for the performance of the proposed TIP algorithm, we now derive the joint Cramèr Rao lower bound (CRLB) to the variance of the UAVs' position and velocity estimates. To this end, we assume that the set of maps $\boldsymbol{\mu}$ is perfectly known, so that it is possible to correctly associate the elements of the lists $\mathcal{D}_{i,j}$ and $\mathcal{V}_{i,j}$ with the corresponding UAVs' identities.

We recall that A UAV out of N are anchors, i.e., their positions and velocities are perfectly known. Therefore, the joint CRLB should only refer to the $\bar{N}{=}N{-}A$ UAVs which are not anchors. Let $\bar{\mathcal{N}}$ be the set of such UAVs and define $\bar{\mathbf{p}}$ and $\bar{\mathbf{v}}$ as the $3\bar{N}$-size vectors obtained by stacking the positions and velocities, $\mathbf{p}_i$ and $\mathbf{v}_i$, respectively, $\forall i{\in}\bar{\mathcal{N}}$. In the CRLB terminology, the position and velocities $\bar{\mathbf{p}}$ and $\bar{\mathbf{v}}$ represent the parameters to be estimated, while the set of distances $\boldsymbol{\mathcal{D}}$ and velocities $\boldsymbol{\mathcal{V}}$ are the observations. Hence, the total number of parameters to estimate is $6\bar{N}$ i.e., 3 position components and 3 velocity components for each UAV in $\bar{\mathcal{N}}$.

Let us first consider the UAV's positions and velocities vectors as random variables, whose components are i.i.d. and have density $f_p(p)$ and $f_v(v)$, respectively. In other words, the joint density of the elements of $\bar{\mathbf{p}}$ is $f_{\bar{\mathbf{p}}}(\bar{\mathbf{p}}){=}\prod_{\ell=1}^{3\bar{N}} f_p(\bar{p}_\ell)$ and, similarly, $f_{\bar{\mathbf{v}}}(\bar{\mathbf{v}}){=}\prod_{\ell=1}^{3\bar{N}} f_v(\bar{v}_\ell)$ where $\bar{p}_\ell$ and $\bar{v}_\ell$ are the ℓ-th elements of $\bar{\mathbf{p}}$ and $\bar{\mathbf{v}}$, respectively.

As already discussed in Sec. V, although the discretization errors $\eta_{i,j,k}$ and $\zeta_{i,j,k}$ appearing in (6) and (7) are functions of (hence, correlated with) $\delta_{i,j,k}$ and $v_{i,j,k}$ (resp.), the uncertainty in the UAVs' positions and velocities can be translated into $\eta_{i,j,k}$ and $\zeta_{i,j,k}$ being i.i.d. random variables, with distributions $f_\eta(\eta)$ and $f_\zeta(\zeta)$ uniform in the ranges $[-\frac{c}{2B}, \frac{c}{2B}]$ and $[-\frac{c}{2f_cT_f}, \frac{c}{2f_cT_f}]$, respectively. Such an assumption allows us to define the joint density of observations and parameters as $f_{\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{V}},\bar{\mathbf{p}},\bar{\mathbf{v}}}(\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{V}},\bar{\mathbf{p}},\bar{\mathbf{v}})$.

It follows that the joint CRLB on the ℓ-th parameter to be estimated is given by $[\mathbf{F}^{-1}]_{\ell,\ell}$ where $\mathbf{F}$ is the $6\bar{N}{\times}6\bar{N}$ Fisher information matrix (FIM) defined as $\mathbf{F}{=}\mathbb{E}_{\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{V}},\bar{\mathbf{p}},\bar{\mathbf{v}}}\left[\mathbf{y}\mathbf{y}^\mathsf{T}\right]$ where

$$\mathbf{y} = \left[\begin{array}{c} \frac{\partial}{\partial \bar{\mathbf{p}}} \log f_{\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{V}},\bar{\mathbf{p}},\bar{\mathbf{v}}}(\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{V}},\bar{\mathbf{p}},\bar{\mathbf{v}}) \\ \frac{\partial}{\partial \bar{\mathbf{v}}} \log f_{\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{V}},\bar{\mathbf{p}},\bar{\mathbf{v}}}(\boldsymbol{\mathcal{D}},\boldsymbol{\mathcal{V}},\bar{\mathbf{p}},\bar{\mathbf{v}}) \end{array}\right]. \tag{33}$$

The derivation of the expression of the FIM is quite tedious, and we only provide the final result, given by:

$$\mathbf{F} = \left[\begin{array}{cc} C_\eta \mathbf{D}_{p,p} + C_\zeta \mathbf{V}_{p,p} + C_p \mathbf{I} & C_\zeta \mathbf{V}_{p,v} \\ C_\zeta \mathbf{V}_{p,v}^\mathsf{T} & C_\zeta \mathbf{V}_{v,v} + C_v \mathbf{I} \end{array}\right]$$

where

$$\begin{aligned}
\mathbf{D}_{p,p} &= \sum_{i,j\neq i,k\neq\{i,j\}} \mathbb{E}_{\bar{\mathbf{p}}}\left[\frac{\partial \delta_{i,j,k}}{\partial \bar{\mathbf{p}}}\frac{\partial \delta_{i,j,k}}{\partial \bar{\mathbf{p}}^\mathsf{T}}\right], \\
\mathbf{V}_{p,p} &= \sum_{i,j\neq i,k\neq i} \mathbb{E}_{\bar{\mathbf{p}},\bar{\mathbf{v}}}\left[\frac{\partial v_{i,j,k}}{\partial \bar{\mathbf{p}}}\frac{\partial v_{i,j,k}}{\partial \bar{\mathbf{p}}^\mathsf{T}}\right], \\
\mathbf{V}_{p,v} &= \sum_{i,j\neq i,k\neq i} \mathbb{E}_{\bar{\mathbf{p}},\bar{\mathbf{v}}}\left[\frac{\partial v_{i,j,k}}{\partial \bar{\mathbf{p}}}\frac{\partial v_{i,j,k}}{\partial \bar{\mathbf{v}}^\mathsf{T}}\right], \\
\mathbf{V}_{v,v} &= \sum_{i,j\neq i,k\neq i} \mathbb{E}_{\bar{\mathbf{p}},\bar{\mathbf{v}}}\left[\frac{\partial v_{i,j,k}}{\partial \bar{\mathbf{v}}}\frac{\partial v_{i,j,k}}{\partial \bar{\mathbf{v}}^\mathsf{T}}\right],
\end{aligned}$$

$$C_x = \int_{-\infty}^{\infty} \frac{(f_x'(z))^2}{f_x(z)}\,\mathrm{d}z\,, \tag{34}$$

with $x{\in}\{p,v,\eta,\zeta\}$. We point out that when $f_x(x)$ is a uniform distribution, the coefficient C_x cannot be computed, due to the discontinuities in its expression. It follows that the joint CRLB cannot be computed under the hypothesis of uniformly distributed $\eta_{i,j,k}$ and $\zeta_{i,j,k}$. To circumvent this problem, and only for the CRLB evaluation, we assume $\eta_{i,j,k}{\sim}\mathcal{N}(0,\sigma_\eta^2)$ and $\zeta_{i,j,k}{\sim}\mathcal{N}(0,\sigma_\zeta^2)$ where $\sigma_\eta{=}\frac{c}{\sqrt{12}B}$ and $\sigma_\zeta{=}\frac{c}{\sqrt{12}f_cT_f}$ so that they have the same variance as their uniformly distributed counterparts. For a Gaussian distribution, the coefficient C_x is given by $C_x{=}1/\sigma_x^2$.

The average joint *per-component* CRLB on the estimate of

the position and velocity vectors are thus given by

$$\overline{\mathrm{CRLB}}_p^{\mathrm{joint}} = \frac{1}{3\bar{N}} \sum_{\ell=1}^{3\bar{N}} \left[\mathbf{F}^{-1}\right]_{\ell,\ell} \tag{35}$$

$$\overline{\mathrm{CRLB}}_v^{\mathrm{joint}} = \frac{1}{3\bar{N}} \sum_{\ell=3\bar{N}+1}^{6\bar{N}} \left[\mathbf{F}^{-1}\right]_{\ell,\ell} . \tag{36}$$

IX. Numerical Results

We now measure the performance of TIP in some test scenarios. As a performance metric, we consider the root mean square error (RMSE) on the position and velocity estimate per UAV and per dimension x, y, z. By recalling that $\bar{\mathcal{N}}$ is the set of non-anchor UAVs, the RMSE on the position estimates is computed by averaging the results obtained from R runs of the TIP algorithm, as

$$\mathrm{RMSE}_p = \sqrt{\frac{1}{3\bar{N}R} \sum_{i\in\bar{\mathcal{N}}} \sum_{r=1}^{R} |\widehat{\mathbf{p}}_{i,r} - \mathbf{p}_{i,r}|^2} \tag{37}$$

where $\mathbf{p}_{i,r}$ is the r-th realization of the positions $\mathbf{p}_i$ and $\widehat{\mathbf{p}}_{i,r}$ is the corresponding estimate. Likewise, the RMSE on the velocity estimates is computed as

$$\mathrm{RMSE}_v = \sqrt{\frac{1}{3\bar{N}R} \sum_{i\in\bar{\mathcal{N}}} \sum_{r=1}^{R} |\widehat{\mathbf{v}}_{i,r} - \mathbf{v}_{i,r}|^2} . \tag{38}$$

As a test scenario, we consider a swarm composed of $\bar{N}$=4 UAVs moving in a flight area and A=4 anchors with fixed positions (their velocity is zero). The N=$\bar{N}$+A UAVs plus anchors communicate among each other using signals with bandwidth B and central frequency f_c=5 GHz. Specifically, we assume that anchors have identities i=1, 2, 3, 4, and their positions, measured in meters, are $\vec{P}_1$=$\vec{0}$, $\vec{P}_2$=1000 $\vec{e}_x$, $\vec{P}_3$=1000 $\vec{e}_y$, and $\vec{P}_4$=1000 $\vec{e}_z$.

In the gradient descent algorithm, we set the stopping threshold to ϵ=10^{-4} and the maximum number of iterations to I_α = 100. For the gradient step γ, we employ the adaptive Barzilai-Borwein method [30], which ensures a faster convergence. Also, we set the threshold $\widetilde{E}_b$=$2E_b$ in (27).

A. Cold start mode

We start by showing the performance of TIP in cold start mode. The results are obtained by averaging the output of TIP over R=100 runs. At each run, the components of the UAV's position vectors were randomly drawn from $\mathcal{N}(\mu, \sigma^2)$ with μ=500 m and σ=$\frac{1{,}000}{\sqrt{12}}$ ≈ 289 m.[6] Also, at each run, TIP is initialized by drawing each component of the tentative positions $\widehat{\mathbf{p}}^{(0)}_{\text{cold start}}$ from the same Gaussian distribution. Likewise, at each run, the components of the UAV velocity vectors were randomly drawn from $\mathcal{N}(0, \sigma^2)$ with σ=10 m/s.

Fig. 4(left) shows the RMSE_p in (37) achieved by TIP, plotted versus the number of iterations of the gradient descend algorithm, I_α, for L=0 (no TIP iterations) and signal bandwidth B=30 MHz, corresponding to a discretization step c/B=10 m. One can notice that I_μ=2 iterations of the BP algorithm provide the same performance as the genie-aided (GA) algorithm, which has perfect knowledge of the maps $\boldsymbol{\mu}$. We recall that GA is used here as a benchmark since it represents a lower bound for the TIP RMSE_p. Fig. 4(left) underlines that, despite the 10 m discretization step in the measurements, the system achieves an RMSE_p of about 1 m after I_α=30 iterations, which means that BP provides very reliable estimates $\widehat{\boldsymbol{\mu}}$. However, for a smaller bandwidth, B=3 MHz, such estimates have much lower reliability: some TIP iterations are then required to improve the RMSE_p, as in the example shown in Fig. 4(right). Here the discretization step is c/B=100 m and the number of BP iterations is I_μ=2. For L=0 (no TIP iterations), the resulting RMSE_p is quite large, about 22 m, while a single TIP iteration lowers it at about 7 m. Importantly, with a further TIP iteration (L=2) and I_α>40 iterations, the system reaches the GA performance.

Fig. 5(left) presents the RMSE_p in (37) versus the signal bandwidth, B, as the number of TIP iterations, L, varies, for I_μ=1. For the sake of comparison, the figure also shows the performance of the GA algorithm, and the joint CRLB_p computed according to (35). We observe that, as B increases, the discretization step decreases and the system provides, in general, more accurate positioning. For L=2, the system performance reaches the GA for all considered values of B

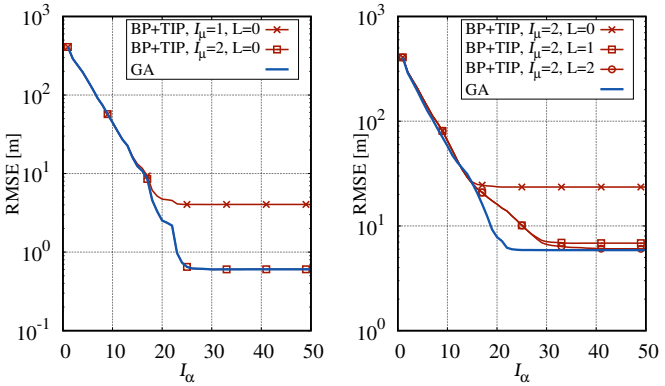

Fig. 4. RMSE as a function of the number of gradient descent iterations (I_α). Left: L=0 and B=30 MHz corresponding to $c\Delta\tau$=10 m. Right: I_μ=2 and B=3 MHz corresponding to $c\Delta\tau$=100 m.

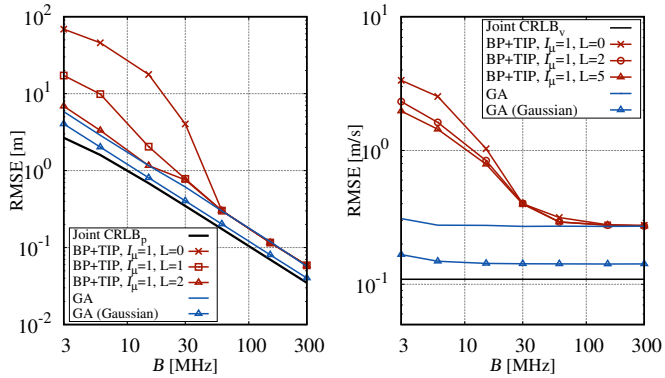

Fig. 5. RMSE on position estimates (left) and velocity estimates (right), as a function of the signal bandwidth, for varying L, T_f=20 ms, and I_μ=1. Average over R=100 random network scenarios.

[6] These are the same μ and σ of a uniform distribution of UAVs in a cube of side 1,000 m with vertices coinciding with the four anchor nodes.

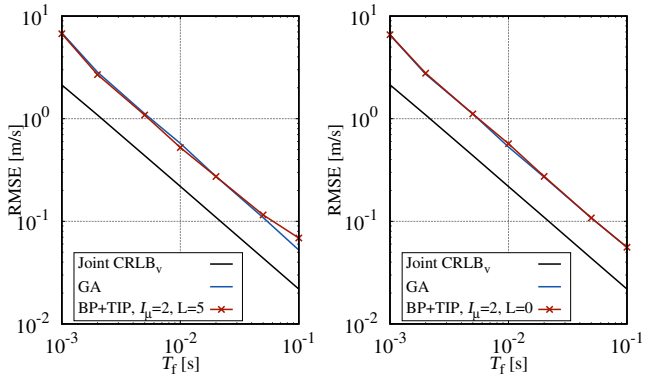

Fig. 6. RMSE on velocity estimation as a function of the frame duration, for $B = 10$ MHz (left) and $B = 30$ MHz (right).

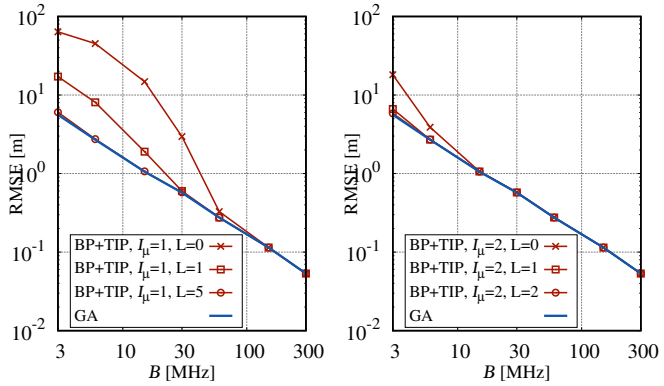

Fig. 7. RMSE on position estimation as a function of the signal bandwidth as L varies, for I_μ=1 (left) and I_μ=2 (right). Average over UAV positions along real-world trajectories from [31].

in the range 3–300 MHz. For the same system parameters and for frame time T_f=20 ms, Fig. 5(right) shows the RMSE_v in (38) on the estimate of the UAV velocities, measured in m/s, plotted versus the signal bandwidth. For such a value of T_f, the discretization step on the velocities is c/f_cT_f=3 m/s. In this setting, the best performance is obtained using L=5 TIP iterations but no further improvement is obtained by increasing L. However, for B=300 MHz, the achieved $\mathrm{RMSE}_v \approx 0.27$ m/s, which is well below the discretization step.

Remark 1 The gap between GA and the CRLB in both plots of Fig. 5 is mainly due to the fact that the quantization errors $\eta_{i,j,k}$ and $\zeta_{i,j,k}$ introduced in (6) and (7), respectively, are due to the rounding operation and depend upon the actual value of $\delta_{i,j,k}$ and $\omega_{i,j,k}$. Instead, the CRLB is computed under the assumption of Gaussian errors $\eta_{i,j,k}$ and $\zeta_{i,j,k}$, which are independent of $\delta_{i,j,k}$ and $\omega_{i,j,k}$, respectively. Indeed, by feeding the TIP algorithm with channel measurements corrupted by Gaussian noise instead of quantization noise, the performance gap is very much reduced, as shown by the curves in Figures 5 (left and right) labeled as "GA (Gaussian)". The remaining small gap is due to the non-optimality of the position/velocity estimation, which in our solution is implemented as two separate blocks, while in an optimal design position and velocity estimation could be implemented jointly at the cost of much higher complexity.

Remark 2 We have observed that the Fisher information matrix (FIM) is dominantly block diagonal for the considered values of the system parameters, and this explains why CRLB_v in Fig. 5(right) is almost independent of the signal bandwidth.

Fig. 6 illustrates the performance of BP+TIP, of the GA algorithm, and of the joint CRLB computed as in (36), for $B = 10$ MHz (left) and $B = 30$ MHz (right). The plots show that the BP+TIP algorithm performs very close to the GA for a range of T_f for any $B \geq 10$ MHz. Similar to Fig. 5, the gap between GA and joint CRLB is due to the fact that quantization noise is not Gaussian and is correlated to the data.

Fig. 7 presents the RMSE on the position estimates, computed by averaging over the set of real UAV positions available in [31], whose coordinates have been linearly scaled to fit in a cube of side 1,000 m centered at $[500, 500, 500]$ m. All other parameters are the same as in Fig. 5, and all curves are also very similar, even though the UAV locations are not random. This demonstrates that our theoretical analysis does apply to real-world scenarios.

B. Tracking mode

To assess the performance of TIP in tracking mode, we consider a test scenario where the UAVs follow trajectories that are deterministic functions of time. As an example of such trajectories, we choose the 3D Lissajous curve [32], which exhibits a simple and easy-to-implement parametric expression. The position vector of UAV i following a Lissajous trajectory is $\vec{P}_i(t) = p_{i,x}(t)\vec{e}_x + p_{i,y}(t)\vec{e}_y + p_{i,z}(t)\vec{e}_z$ whose components are given by: $p_{i,s}(t)=a_{i,s}\sin(b_{i,s}t + \phi_{i,s})$ where $a_{i,s}$, $b_{i,s}$, and $\phi_{i,s}$ are parameters and $s \in \{x, y, z\}$. As a consequence, the instantaneous velocity vector of the i-th UAV is given by $\vec{V}_i(t)=v_{i,x}(t)\vec{e}_x + v_{i,y}(t)\vec{e}_y + v_{i,z}(t)\vec{e}_z$ with components

$$v_{i,s}(t) = \frac{\mathrm{d}p_{i,s}(t)}{\mathrm{d}t} = a_{i,s}b_{i,s}\cos(b_{i,s}t + \phi_{i,s}) \qquad (39)$$

for $s \in \{x, y, z\}$.

Fig. 8 depicts a 2D projection of a portion of the same 3D Lissajous trajectory. This is an instance of a random trajectory obtained by independently drawing the parameters $a_{i,s}$ from the uniform distribution $\mathcal{U}[0, 1]$, $b_{i,s}$ from $\mathcal{U}[0, 0.2]$, and $\phi_{i,\ell}$ from $\mathcal{U}[0, 2\pi]$. The signal bandwidth is set to B=3 MHz and B=300 MHz, in Fig. 8(left) and Fig. 8(right), respectively. In both figures, T_f=20 ms, and the update time is Δt=1 s. In the plots, the UAV trajectory is represented by the solid black line, while blue circles refer to the UAV position, $\vec{P}_i(t)$ at $t=n\Delta t$, $n=0, 1, \ldots, 50$. The red circles denote the positions $\widehat{\vec{P}}_i(n\Delta t)$ estimated by TIP for L=5 and the arrows represent the estimated velocity vectors $\widehat{\vec{V}}_i(n\Delta t)$. The actual velocity vectors $\vec{V}_i(n\Delta t)$ (not shown) are tangent to the black line at $t=n\Delta t$. The trajectory is traveled by the UAV from right to left and the blue circles mark 1 s-time intervals. For B=3 MHz, the positions are fairly well estimated (i.e., the red circles are always close to the blue ones). However, errors in estimating the maps sometimes result in a poor estimate of the velocity vectors, in both direction and magnitude. By increasing the bandwidth to 300 MHz, the map estimates become very reliable: position estimates are excellent and the velocity vectors perfectly follow the tangent to the trajectory.

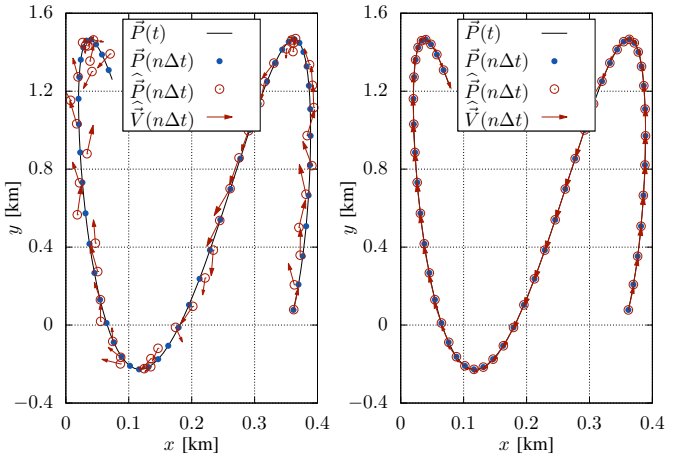

Fig. 8. 2D projection of a portion of a UAV Lissajou trajectory, for N=8, A=4, B=3 MHz (left) and B=300 MHz (right), T_f=20 ms, and Δt=1 s. The solid black line represents the trajectory, while blue circles indicate the actual UAV positions, $\vec{P}_i(n\Delta t)$. The red circles denote the estimated positions $\widehat{\vec{P}}_i(n\Delta t)$, and the arrows represent the estimated velocities $\widehat{\vec{V}}_i(t)$.

X. Conclusions

We proposed an iterative algorithm, named TIP, for localizing and tracking the position of UAVs communicating with each other, by using RED measurements obtained through OTFS-modulated signals. TIP exploits belief propagation and gradient descent optimization, to achieve precise positions and velocities of the UAVs even when OTFS channel estimation accuracy is limited because of the low resolution of the delay and Doppler shift grid. Our results show a 10-fold improvement of the RMSE over the maximum delay discretization error ($c/2B$) in the estimate of the length of the position vector, which translates to a 3- to 4-fold improvement on each 3D component. Taking advantage of the measurements from all the communication links between the UAVs, the remarkable advantage of our solution is thus the improvement of the c/B resolution of a multi-target radar system with a limited bandwidth B. As no other algorithm using delay-Doppler channel measurements is available in the literature, we demonstrated the excellent performance of TIP against the Cramér Rao lower bound and a genie-aided version of TIP (with perfect knowledge of the maps). Future work will extend our method to include passive reflectors for terrestrial applications.

In this work, we have assumed the pilot power to be sufficient to guarantee that all paths are resolved, i.e., that all lists $\mathcal{D}_{i,j}$ and $\mathcal{V}_{i,j}$ contain $N-2$ nLoS paths. The case where lists are incomplete, e.g., due to some paths being too weak, is a direction we are currently pursuing, although, determining the UAV's identities in the incomplete lists is very challenging.

Appendix A
Derivation of (4)

Let $\vec{P}_i$ be the position vector of UAV i at time $t=0$ and $\vec{V}_i$ its instantaneous velocity vector. Then, for small t, the position of i can be written as $\vec{P}_i(t)=\vec{P}_i+t\vec{V}_i+o(t^2)$ or, in terms of their components, $\mathbf{p}_i(t)=\mathbf{p}_i+t\mathbf{v}_i+o(t^2)$.

The velocity of UAV j, as observed by i through the reflexion on k, is the derivative w.r.t. time (evaluated at t=0) of the distance $s_{i,j,k}(t)$ between j and i through k. In turn, this distance is the sum of two contributions: the distance between j and k, and the distance between k and i. Hence,

$$s_{i,j,k}(t)=|\mathbf{p}_j(t)-\mathbf{p}_k(t)|+|\mathbf{p}_k(t)-\mathbf{p}_i(t)|\,.$$

By using the chain rule, for any n and m, we get

$$\left.\frac{\mathrm{d}|\mathbf{p}_n(t)-\mathbf{p}_m(t)|}{\mathrm{d}t}\right|_{t=0}=(\mathbf{v}_n-\mathbf{v}_m)^{\mathsf{T}}\frac{\mathbf{p}_n-\mathbf{p}_m}{|\mathbf{p}_n-\mathbf{p}_m|}\,. \tag{40}$$

It follows that the expression for $\omega_{i,j,k}$ is given by (4).

References

[1] A. Nordio, C. F. Chiasserini, and E. Viterbo, "Robust localization of UAVs in OTFS-based networks," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 7471–7477.

[2] K. Witrisal, P. Meissner, E. Leitinger, Y. Shen, C. Gustafson, F. Tufvesson, K. Haneda, D. Dardari, A. F. Molisch, A. Conti, and M. Z. Win, "High-accuracy localization for assisted living: 5G systems will turn multipath channels from foe to friend," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 59–70, 2016.

[3] A. R. Ansari, N. Saeed, M. I. Ul Haq, and S. Cho, "Accurate 3D localization method for public safety applications in vehicular ad-hoc networks," *IEEE Access*, vol. 6, pp. 20 756–20 763, 2018.

[4] S. M. Asaad and H. S. Maghdid, "A comprehensive review of indoor/outdoor localization solutions in IoT era: Research challenges and future perspectives," *Computer Networks*, vol. 212, p. 109041, 2022.

[5] Y. Hong, T. Thaj, and E. Viterbo, *Delay-Doppler Communications: Principles and Applications*. Academic Press, Elsevier, 2022.

[6] P. Raviteja, K. T. Phan, Y. Hong, and E. Viterbo, "Orthogonal time frequency space (OTFS) modulation based radar system," in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–6.

[7] L. Han, R. Liu, X. Lv, Z. Wang, and Q. Zhu, "Optimal waveform design for integrated localization and communication in OFDM systems," in *2022 IEEE 8th International Conference on Computer and Communications (ICCC)*, 2022, pp. 868–874.

[8] J. Bai, G. Wei, S. Wang, X. Wang, and Z. Fei, "Efficient direct localization of OFDM emitters in multipath environment with mobile receivers," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 545–556, 2022.

[9] M. J. Emadi, S. Hu, and H. Wang, "Precise positioning: When OTFS meets distributed cooperative positioning," Feb. 2023. [Online]. Available: 10.36227/techrxiv.21951014.v1

[10] F. Linsalata, A. Albanese, V. Sciancalepore, F. Roveda, M. Magarini, and X. Costa-Perez, "OTFS-superimposed PRACH-aided localization for UAV safety applications," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[11] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[12] B. Etzlinger, F. Meyer, F. Hlawatsch, A. Springer, and H. Wymeersch, "Cooperative simultaneous localization and synchronization in mobile agent networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 14, pp. 3587–3602, 2017.

[13] J. M. Olson, N. A. Toombs, and T. W. McLain, "Multi-agent mapping and navigation of unknown GPS-denied environments using a relative navigation framework," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 1221–1230.

[14] L. Lu, J. Hu, Q. Pan, C. Zhao, Z. Xu, and C. Jia, "Decentralized localization for multi-agent systems based on asynchronous communication," in *2021 40th Chinese Control Conference (CCC)*, 2021, pp. 5489–5494.

[15] M. Borelle, S. Bertrand, C. Stoica, T. Alamo, and E. F. Camacho, "Cooperative localization of an UAV fleet using distributed MHE with EKF pre-estimation and nonlinear measurements," in *2023 27th International Conference on System Theory, Control and Computing (ICSTCC)*, 2023, pp. 143–148.

[16] R. Wang, C. Xu, J. Sun, S. Duan, and X. Zhang, "Cooperative localization for multi-agents based on reinforcement learning compensated filter," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 10, pp. 2820–2831, 2024.

[17] H. Zhao, M. Huang, and Y. Shen, "High-accuracy localization in multipath environments via spatio-temporal feature tensorization," *IEEE Transactions on Wireless Communications*, vol. 21, no. 12, pp. 10 576–10 591, 2022.

[18] L. Wielandner, E. Leitinger, F. Meyer, and K. Witrisal, "Message passing-based 9-D cooperative localization and navigation with embedded particle flow," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 9, pp. 95–109, 2023.

[19] S. Aditya, A. F. Molisch, N. Rabeah, and H. M. Behairy, "Localization of multiple targets with identical radar signatures in multipath environments with correlated blocking," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 606–618, 2018.

[20] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[21] T. Kirubarajan and Y. Bar-Shalom, "Probabilistic data association techniques for target tracking in clutter," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 536–557, 2004.

[22] S. He, H.-S. Shin, and A. Tsourdos, "Joint probabilistic data association filter with unknown detection probability and clutter rate," in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2017, pp. 559–564.

[23] F. Liu, Y. Cui, C. Masouros, J. Xu, T. X. Han, Y. C. Eldar, and S. Buzzi, "Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 6, pp. 1728–1767, 2022.

[24] Z. Sun, Z. Yu, B. Guo, B. Yang, Y. Zhang, and D. W. K. Ng, "Integrated sensing and communication for effective multi-agent cooperation systems," *IEEE Communications Magazine*, vol. 62, no. 9, pp. 68–73, 2024.

[25] F. Meyer, T. Kropfreiter, J. L. Williams, R. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 221–259, 2018.

[26] E. Leitinger, F. Meyer, F. Hlawatsch, K. Witrisal, F. Tufvesson, and M. Z. Win, "A belief propagation algorithm for multipath-based SLAM," *IEEE Transactions on Wireless Communications*, vol. 18, no. 12, pp. 5613–5629, 2019.

[27] F. Meyer, A. Tesei, and M. Z. Win, "Localization of multiple sources using time-difference of arrival measurements," in *ICASSP'17*, 2017, pp. 3151–3155.

[28] Y. Wang and K. C. Ho, "TDoA source localization in the presence of synchronization clock bias and sensor position errors," *IEEE Transactions on Signal Processing*, vol. 61, no. 18, pp. 4532–4544, 2013.

[29] P. Raviteja, K. T. Phan, and Y. Hong, "Embedded pilot-aided channel estimation for OTFS in delay–doppler channels," *IEEE Trans. Veh. Tech.*, vol. 68, no. 5, pp. 4906–4917, 2019.

[30] J. Barzilai and J. M. Borwein, "Two-Point Step Size Gradient Methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 01 1988. [Online]. Available: https://doi.org/10.1093/imanum/8.1.141

[31] "Drone tracking dataset," https://github.com/CenekAlbl/drone-tracking-datasets.

[32] M. G. V. Bogle, J. E. Hearst, V. F. R. Jones, and L. Stoilov, "Lissajous knots," *Journal of Knot Theory and Its Ramifications*, vol. 03, no. 02, pp. 121–140, 1994.