

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/344098126>

3D Map-based Trajectory Design in UAV-aided Wireless Localization Systems

Preprint · September 2020

CITATIONS

0

READS

15

3 authors, including:



Omid Esrafilian

EURECOM

16 PUBLICATIONS 131 CITATIONS

[SEE PROFILE](#)



Rajeev Gangula

EURECOM

21 PUBLICATIONS 122 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Autonomous Robotics [View project](#)

3D Map-based Trajectory Design in UAV-aided Wireless Localization Systems

Omid Esrafilian, Rajeev Gangula, and David Gesbert

Abstract—This paper considers the problem of localizing outdoor ground radio users with the help of an unmanned aerial vehicle (UAV) on the basis of received signal strength (RSS) measurements in an urban environment. We assume that the propagation model parameters are not known a priori, and depending on the UAV location, the UAV-user link can experience either line-of-sight (LoS) or non-line-of-sight (NLoS) propagation condition. We assume that a 3D map of the environment is available which the UAV can exploit in the localization process. Based on the proposed map-aided estimator, we devise an optimal UAV trajectory to accelerate the learning process under a limited mission time. To do so, we borrow tools such as Fisher information from the theory of optimal experiment design. Our map-aided estimator achieves superior localization accuracy compared to the map-unaware methods, and our simulations show that optimized UAV trajectory achieves superior learning performance compared to random trajectories.

Index Terms—UAV, drone, trajectory design, learning, localization, 3D map.

I. INTRODUCTION

FINDING geographic location of user devices or nodes from the radio signals emitted by them has found many applications in wireless sensor networks [1], navigation and tracking [2], location based services [3], etc. In a typical wireless network, nodes with perfectly known positions, called anchor nodes, which can be stationary or mobile, are used to collect measurements from the users whose location need to be estimated. Different types of measurements such as received signal strength (RSS), time-of-arrival (TOA), angle of arrival (AOA), etc., can be obtained by the anchor nodes for user localization [4], [5].

While elements of the terrestrial network such as access points and base stations (BSs) are often used as anchor nodes, recent advancements in wireless communication and robotic technologies have made it possible to have flying radio infrastructure where commercial grade unmanned aerial vehicles (UAVs) can be integrated with small BSs or relays [6] and access points (WiFi, Bluetooth, etc.) [7]. Such UAVs can also be used as anchor nodes, thanks to their inherent advantages in terms of 3D mobility and ultra-flexible deployment. Using UAVs as aerial anchors to localize the ground users has gained interest recently [8]–[13]. Further, the UAV can

localize radio nodes on-the-go, while attending other missions where positioning and tracking are beneficial, for example precision delivery, emergency response systems, optimizing the trajectories of UAV BSs or relays in flying radio access networks (FRANs) [14]–[16], wild life tracking [17], etc.

When it comes to the type of measurements exploited for localization, RSS measurements offers an attractive option as they are easy to obtain in many wireless networks. Unlike the timing based methods, they do not require tight synchronization and calibration of the transceivers. The problem of network localization based on RSS measurements obtained by static anchor nodes with unknown propagation pathloss model parameters has been studied before [18]–[20]. The work in [18] uses a pathloss model that does not differentiate between line-of-sight (LoS) and non-line-of-sight (NLoS) conditions. Since this paper is targeting localization with the help of UAV in arbitrary environments including challenging urban scenarios, the air-to-ground propagation between the UAV and ground users is expected to follow a segmented model where the pathloss condition will rapidly switch between LoS and NLoS depending on the UAV location [21], [22]. The authors in [19], [20] extended the work of [18] to the above segmented pathloss model with [19] modelling the collected RSS measurements as mixture of two Gaussian distributions with weights representing the probability of the measurement obtained from LoS or NLoS segment, and the work in [20] tries to do hard classify measurements into LoS and NLoS categories. The problem of joint user localization and pathloss parameter estimation is then solved iteratively using an expectation and maximization (EM) criterion in [19], and by an iterative classification and least square approach in [20]. In this work, we however point out that, if a 3D city map containing the building structure information is available (map can be obtained offline for example Google map or it can be constructed online using photogrammetry techniques [23] or from the radio measurements [24]), it can be leveraged to enhance the LoS/NLoS classification of a measurement. Exploiting such information may lead to better classification of measurements into LoS and NLoS segments unlike the pure statistical approaches in [19], [20] which ignore the map information. In different contexts, fingerprinting-based localization systems that implicitly exploits the map [25] and building *map-aware* statistical models [26] have shown to improve the localization performance. However, the methods in [25], [26] require an extensive prior measurement campaign in the area where the network is deployed.

When using UAV as a mobile anchor in a network for localization, a central challenge lies in the ability to localize

This work was supported by the ERC under the European Union Horizon 2020 research and innovation program (Agreement no. 670896).

The authors are with the Department of Communication Systems, EURECOM, Sophia-Antipolis, France (email: {esrafilian, gangula, gesbert}@eurecom.fr).

Copyright (c) 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

the radio nodes with enough accuracy within the limited flying time constrained by the on-board limited energy capacity battery. This in turn raises the question of *where* should the UAV fly so as to collect the most informative measurements from a user localization perspective. This problem has also received some attention in the literature. In [8], the UAV altitude is optimized to minimize the localization accuracy of outdoor users in an urban environment by considering both LoS and NLoS RSS measurements. The authors in [9] studied the problem of single-user localization from the RSS measurements by assuming that the collected measurements by the UAV fall into the LoS category. An online trajectory for the UAV is also designed that keeps the UAV heading towards the estimated user position. The authors in [10], focused on the experimental exploration of static trajectories applied to the localization of wireless nodes using UAVs. They investigated different types of trajectories (so-called Triangle and Circle trajectories) and compared the performance of localization by following these trajectories in their indoor-outdoor settings. In [11]–[13] the problem of trajectory design of multiple flying anchors (UAVs) for RSS-based node localization is investigated. Authors in [11]–[13] assumed that the UAVs fly high enough so that they can establish LoS links to any nodes in the network at all times, and the effect of NLoS measurements is ignored in the localization process which is not very realistic for mid-altitude UAVs.

In this work, we devise a UAV trajectory that optimizes the learning performance in user localization under a given mission time. To the best of our knowledge, UAV-based user localization exploiting a realistic segmented pathloss model fed from a 3D map, and optimizing learning trajectories in such context has not been studied before. Specifically, our contributions are as follows:

- We formulate and solve the problem of map-based joint pathloss parameter estimation and user localization from the RSS measurements collected by the UAV from ground users in both LoS and NLoS conditions¹. The proposed algorithm is based on particle swarm optimization (PSO) aided with the 3D map of the environment. We also prove the convergence of the proposed algorithm.
- We then proceed to design a trajectory for the UAV to optimize the performance of localization and channel learning under a maximum mission duration constraint. To achieve this, we propose an on-line algorithm to find the optimal trajectory where we borrow tools from optimal experiment design where metrics like Fisher information is used to find most informative measurements from a user localization perspective.

The rest of the paper is organized as follows: Section II introduces the system model. In Section III, we formulate and solve the problem of learning the channel parameters and localizing the users on the basis of an arbitrary (non optimized) trajectory. In Section IV, we formulate and solve the trajectory optimization problem. Numerical results are

¹This work can be extended to fuse different measurements such as RSS, TOA, etc.

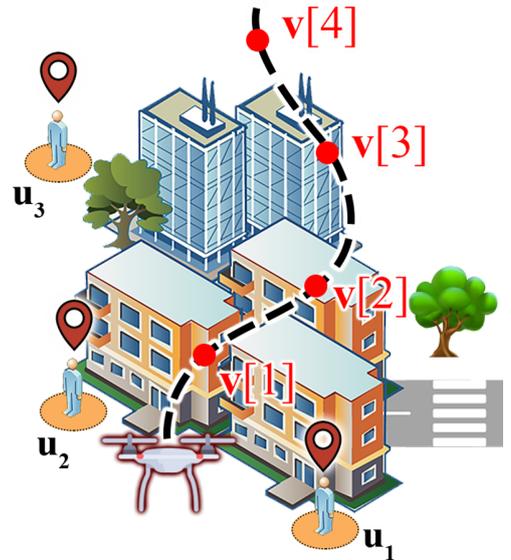


Fig. 1: UAV-aided ground user localization system.

presented in Section V to validate the performance of the proposed algorithms. Finally, Section VI concludes the paper.

Notation: Matrices are represented by uppercase bold letters, vectors are represented by lowercase bold letters. Sets are indicated by calligraphic uppercase letters. The transpose of matrix \mathbf{A} is denoted by \mathbf{A}^T . The trace of matrix \mathbf{A} are denoted by $\text{tr}(\mathbf{A})$. The expectation operator is denoted by $E[\cdot]$. The set of integers from m to n , $m < n$, is represented by $[m, n]$. The Euclidean norm of vector \mathbf{a} is denoted by $\|\mathbf{a}\|$, and $|a|$ denotes the absolute value of scalar a . Note that the cardinality of set \mathcal{A} is also denoted by $|\mathcal{A}|$.

II. SYSTEM MODEL

We consider a scenario similar to the one depicted in Fig. 1, where a UAV-mounted access point or base station that is connected to K ground level users in an urban area consisting of a number of city buildings. The users are scattered all over the city and $\mathbf{u}_k = [x_k, y_k]^T \in \mathbb{R}^2$, $k \in [1, K]$ denotes the k -th user's location. The users are considered static and their locations are unknown. The UAV's localization mission lasts for a duration T , during which the aim of the UAV is to estimate the unknown user locations based on RSS measurements from them. Note that a constraint on flying time reflects the limited on-board battery capacity of the UAV. However, a more advanced power consumption models of the UAV that takes into account parameters such as acceleration, hovering etc. as in [27] can also be considered in trajectory design which is left for the future work. A 3D map of the environment where the UAV and users are located is assumed to be available.

We assume that the time period $[0, T]$ is discretized into N equal length intervals, each of duration $\delta = T/N$, indexed by $n = 1, \dots, N$. The value of δ is chosen to be sufficiently small such that UAV's location, velocity, and heading angles can be considered constant in one interval. In the n -th interval, the UAV/drone position is denoted by $\mathbf{v}[n] = [x[n], y[n], z[n]]^T \in$

\mathbb{R}^3 . We assume that the drone is equipped with a GPS receiver, hence $\mathbf{v}[n], \forall n$ is known. During the mission, drone's position evolves according to

$$\mathbf{v}[n+1] = \mathbf{v}[n] + \begin{bmatrix} \cos(\phi[n]) \cos(\psi[n]) \\ \sin(\phi[n]) \cos(\psi[n]) \\ \sin(\psi[n]) \end{bmatrix} \rho[n], \quad (1a)$$

$$0 \leq \rho[n] \leq \rho_{max}, n \in [1, N-1], \quad (1b)$$

$$0 \leq \phi[n] \leq 2\pi, n \in [1, N-1], \quad (1c)$$

$$-\frac{\pi}{2} \leq \psi[n] \leq \frac{\pi}{2}, n \in [1, N-1], \quad (1d)$$

$$h_{min} \leq z[n] \leq h_{max}, n \in [1, N-1], \quad (1e)$$

where in the n -th time slot, $\rho[n]$ represents the distance traveled by the drone, $\phi[n]$ and $\psi[n]$ represent the heading and elevation angles, respectively. The maximum distance traveled in a time slot is denoted by ρ_{max} and it depends on the maximum velocity. The constraint (1e) reflects the fact that the drone always flies at an altitude higher than h_{min} and lower than h_{max} , with h_{min} being the height of the tallest building in the city.

III. USER LOCALIZATION WITH AN ARBITRARY TRAJECTORY

In this section, we propose a map-based algorithm to estimate the user locations from the channel gain measurements collected by the UAV. Here UAV follows an arbitrary trajectory for collecting the measurements, while the problem of trajectory optimization is deferred to section IV. The measurement collection process from the users is described next followed by the estimation problem.

Let us denote an arbitrary trajectory taken by the UAV during the mission by a sequence $\chi = \{\mathbf{v}[n], n \in [1, N]\}$, where $\mathbf{v}[n]$ represents the UAV's position in the n -th time interval. We assume that the UAV flies over N different locations during the mission. From each of these locations, the UAV collects radio measurements from all K users. Let $g_{n,k}$ represent the channel gain or RSS measurement (in dB scale) obtained from the k -th user by the UAV in the n -th interval. Using the segmented pathloss model that is suitable for air-to-ground channels in urban environments with buildings [21], [22], we have

$$g_{n,k} = \begin{cases} \lambda_n(\boldsymbol{\theta}_{LoS}, \mathbf{u}_k) + \eta_{n,k,LoS} & \text{if LoS} \\ \lambda_n(\boldsymbol{\theta}_{NLoS}, \mathbf{u}_k) + \eta_{n,k,NLoS} & \text{if NLoS} \end{cases}, \quad (2)$$

where

$$\lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k) = \beta_s - \alpha_s 10 \log_{10}(\|\mathbf{u}_k - \mathbf{v}[n]\|), \quad (3)$$

$\boldsymbol{\theta}_s = [\alpha_s, \beta_s]^T, s \in \{\text{LoS}, \text{NLoS}\}$, α_s is the pathloss exponent, β_s is the channel gain offset, and $\eta_{n,k,s}$ represents shadowing effect² with zero-mean Gaussian distribution with variance σ_s^2 . We assume that NLoS measurements have a higher shadowing affect i.e, $\sigma_{NLoS} \geq \sigma_{LoS}$ [28], and the values of

$\sigma_s^2, s \in \{\text{LoS}, \text{NLoS}\}$ are known. The probability distribution of the measurement in (2) can be modeled as

$$p(g_{n,k}) = (f_{n,k,LoS})^{\omega_{n,k}} (f_{n,k,NLoS})^{(1-\omega_{n,k})}, \quad (4)$$

where

$$f_{n,k,s} = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{(g_{n,k} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k))^2}{2\sigma_s^2}\right),$$

and $\omega_{n,k} \in \{0, 1\}$ is the classifier binary variable (yet unknown) indicating whether a measurement falls into the LoS or NLoS category.

Assuming that collected measurements conditioned on channel parameters³ and user positions are independent and identically distributed (i.i.d), using (4), the maximum likelihood estimation (MLE) of $\boldsymbol{\theta}_s$ and \mathbf{u}_k leads to minimizing

$$\begin{aligned} \mathcal{L} = \log\left(\frac{\sigma_{LoS}^2}{\sigma_{NLoS}^2}\right) & \sum_{k=1}^K \sum_{n=1}^N \omega_{n,k} + \\ & \sum_{k=1}^K \sum_{n=1}^N \frac{\omega_{n,k}}{\sigma_{LoS}^2} |g_{n,k} - \lambda_n(\boldsymbol{\theta}_{LoS}, \mathbf{u}_k)|^2 + \\ & \sum_{k=1}^K \sum_{n=1}^N \frac{(1-\omega_{n,k})}{\sigma_{NLoS}^2} |g_{n,k} - \lambda_n(\boldsymbol{\theta}_{NLoS}, \mathbf{u}_k)|^2, \end{aligned} \quad (5)$$

The estimate of $\boldsymbol{\theta}_s$ and \mathbf{u}_k can then be obtained by solving

$$\min_{\substack{\omega_{n,k}, \mathbf{u}_k, \forall n, \forall k \\ \boldsymbol{\theta}_{LoS}, \boldsymbol{\theta}_{NLoS}}} \mathcal{L} \quad (6a)$$

$$\text{s.t. } \omega_{n,k} \in \{0, 1\}, \forall n, \forall k. \quad (6b)$$

Since the objective function in (6) comprises of binary variables $\omega_{n,k}$, and the fact that (3) is a non-linear and non-convex function of the user location \mathbf{u}_k , it is challenging to solve the simultaneous classification, localization and channel learning problem in (6) optimally. To tackle this problem, we use a PSO algorithm. As will be clear later, the PSO algorithm is enhanced to exploit the side information stemming from the 3D map of the environment. Before presenting the solution to (6), we give a brief introduction to the PSO algorithm.

A. Particle Swarm Optimization

PSO [29] is a computational method that tries to find the solution to an optimization problem by iteratively trying to improve a candidate solution with regard to a given measure of quality, so called fitness value. Initially, a population of random candidate solutions, called particles, are generated and then in each iteration these particles are moved around their neighborhood across the search-space based on a simple mathematical formulae capturing each particle's position and so-called velocity, as shown later in (7). After each iteration, each particle is evaluated by the objective function, determining the fitness of that particle. Particles are generated in a swarm of size C , where the j -th particle is denoted as

²The shadowing component can also be used to incorporate the additional noise terms such as UAV's GPS error, etc.

³This amounts to assuming the shadowing coefficients are independent over successive UAV locations, which is a classical simplifying assumption, see for e.g. [22]

$\mathbf{c}_j \in \mathbb{R}^D$, $j = [1, C]$, and the particle's dimension D is equal to the number of unknown parameters to be estimated. The fitness and the velocity of each particle need to be stored. Critically, each particle remembers the best fitness value it has achieved so far during the operation of the algorithm, referred to as the *individual best fitness*, as well as the candidate solution that achieved this fitness, referred to as the *individual best candidate solution*. Finally, the PSO algorithm maintains the best fitness value achieved among all particles in the population, called the *global best fitness*, as well as the candidate solution that achieved this fitness, called the *global best candidate solution*.

At each iteration, the velocity of each particle in the swarm is updated using the following equation:

$$\dot{\mathbf{c}}_j^{(i+1)} = \nu \dot{\mathbf{c}}_j^{(i)} + \varepsilon_1 r_1 (\mathbf{c}_j^b - \mathbf{c}_j^{(i)}) + \varepsilon_2 r_2 (\mathbf{c}^* - \mathbf{c}_j^{(i)}), \quad (7)$$

where $\mathbf{c}_j^{(i)}$, $\dot{\mathbf{c}}_j^{(i)}$ are the j -th particle's position and its velocity at the i -th iteration of PSO, respectively. The individual best candidate solution for particle j at each iteration is denoted by \mathbf{c}_j^b , and \mathbf{c}^* represents the swarm's global best candidate solution. The parameters ν, ε_1 and ε_2 are user-supplied coefficients. ν is an inertial coefficient which can either dampen the particle's inertia or accelerate the particle in its original direction. The cognitive coefficient ε_1 , affects the size of the step the particle takes toward its individual best candidate solution, and ε_2 is the social coefficient which modulates the step the particle takes toward the global best candidate solution so far. The values r_1, r_2 ($0 \leq r_1 \leq 1$, $0 \leq r_2 \leq 1$) are random values regenerated from a uniform distribution for each velocity update. Once the velocity for each particle is calculated, each particle's position is updated by applying the new velocity to the particle's previous position:

$$\mathbf{c}_j^{(i+1)} = \mathbf{c}_j^{(i)} + \dot{\mathbf{c}}_j^{(i+1)}. \quad (8)$$

This process is repeated until some stopping condition is met (e.g. a preset number of iterations of the PSO algorithm, a predefined target fitness value, etc.). Since PSO algorithm works by jointly maintaining several candidate solutions in the search area, it is less likely to be trapped into a local minimum which often arises in non-convex optimization as opposed to the gradient-based optimization methods.

B. Single User Case

As explained in Section III-A, PSO method can be useful in solving variety of non-convex optimization problems and is especially appropriate here due to the non-linear and non-convex structure of problem (6). Moreover, using PSO allows us to exploit the 3D map information to obtain a more precise estimate of the user location, which is explained later in this section. If we blindly apply the PSO method to solve (6), the particle needs to be defined as

$$\mathbf{c}_j = [w_{1,1}, \dots, w_{N,K}, \mathbf{u}_1^T, \dots, \mathbf{u}_K^T, \boldsymbol{\theta}_{\text{LoS}}^T, \boldsymbol{\theta}_{\text{NLoS}}^T]^T, j \in [1, C], \quad (9)$$

where C denotes the number of particles, and each particle consists of $K \times N$ classification variables, $2K$ user locations variables (in two dimensions), and four unknown variables

$\alpha_s, \beta_s, s \in \{\text{LoS}, \text{NLoS}\}$. However, with the number of measurements being very large, applying PSO with such particles defined in (9) incurs very high complexity in the algorithm as we need to maintain and update several particles of very large dimensions during the run time. To solve this difficulty, we propose to exploit the 3D map to reduce the complexity of PSO algorithm by replacing the particle \mathbf{c}_j as defined in (9) which is of very large dimension with the particle now defined as

$$\mathbf{c}_j = [\mathbf{u}_1^T, \dots, \mathbf{u}_K^T]^T \in \mathbb{R}^{2K}, \forall j. \quad (10)$$

The motivation for the above is that, propagation segment classification variables $w_{n,k}$ can in fact be directly inferred from the UAV and user locations (particle) from a trivial geometry argument: For a given UAV position, the user is considered in LoS to the UAV if the straight line passing through the UAV's and the user position lies higher than any buildings in between. In turn, having classified each measurement into LoS or NLoS, the channel parameters α_s and β_s can be learned easily as well from the measurements by resorting to a standard least square (LS) technique. As a result, the pathloss parameters estimates can be removed from the particles. We now describe the map-aided PSO algorithm in a more detailed manner, we start with a single user scenario and later extend it to the multi-user case.

In a single user case ($K = 1$), each particle has a dimension of $\mathbf{c}_j := \mathbf{u}_1 \in \mathbb{R}^2$ (i.e. each particle is a potential candidate for the user location). Therefore, the likelihood (5) for a given particle, which is an estimate of the user location, is given by

$$\mathcal{L}(\mathbf{c}_j^{(i)}) = \log \left(\frac{\sigma_{\text{LoS}}^2}{\sigma_{\text{NLoS}}^2} \right) |\mathcal{M}_{\text{LoS},1,j}| + \sum_{s \in \{\text{LoS}, \text{NLoS}\}} \sum_{n \in \mathcal{M}_{s,1,j}} \frac{1}{\sigma_s^2} |g_{n,1} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{c}_j^{(i)})|^2, \quad (11)$$

where $\mathbf{c}_j^{(i)}$ is the j -th particle at the i -th iteration of the PSO algorithm, and $\mathcal{M}_{s,1,j}$ is a set of time indices of measurements collected from user 1 which are in segment s by assuming that the user 1 location is same as particle j . To form $\mathcal{M}_{s,1,j}$, a 3D map of the city is utilized. For example, measurement $g_{n,1}$ is considered LoS, if the straight line passing through $\mathbf{c}_j^{(i)}$ and the drone location $\mathbf{v}[n]$ lies higher than any buildings in between. An illustration of the map-aided classification by considering merely two particles is shown in Fig. 2.

Having formed $\mathcal{M}_{s,1,j}$, for the given particle $\mathbf{c}_j^{(i)}$, (11) is minimized just by optimizing over the channel parameters $\boldsymbol{\theta}_{\text{LoS}}, \boldsymbol{\theta}_{\text{NLoS}}$ as follows

$$\mathcal{L}^*(\mathbf{c}_j^{(i)}) := \min_{\boldsymbol{\theta}_{\text{LoS}}, \boldsymbol{\theta}_{\text{NLoS}}} \mathcal{L}(\mathbf{c}_j^{(i)}). \quad (12)$$

The problem in (12) is a standard LS problem and can be easily solved by using well known methods [30].

We denote the index of the best particle minimizing (11) as

$$j^* := \arg \min_{j \in [1, C]} \mathcal{L}^*(\mathbf{c}_j^{(i)}). \quad (13)$$

Consequently, the individual best fitness and global best fitness

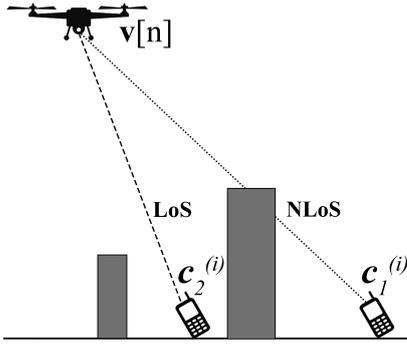


Fig. 2: Map-aided LoS/NLoS classification of two particles.

values are, respectively, updated as follows

$$\begin{aligned} F_j &:= \min(\mathcal{L}^*(\mathbf{c}_j^{(i)}), F_j), j \in [1, C], \\ F^* &:= \min(\mathcal{L}^*(\mathbf{c}_{j^*}^{(i)}), F^*). \end{aligned} \quad (14)$$

The individual best candidate solution for particle j at each iteration (\mathbf{c}_j^b), and the swarm's global best candidate solution (\mathbf{c}^*) are also updated accordingly. In addition, we denote the learned channel parameters corresponding to \mathbf{c}^* by $\theta_{\text{LoS}}^*, \theta_{\text{NLoS}}^*$. The PSO algorithm then proceeds to the next iteration by updating the particles following (7) and (8). We assume that the PSO terminates after I iterations.

In accordance with (14), F^* is non-increasing at each iteration, therefore the original problem (6) can be upper-bounded as:

$$\mathcal{L}^* \leq F^*, \quad (15)$$

where \mathcal{L}^* is the global minimum value of (6). By assuming a large number of particles, \mathcal{L}^* can be approximated by the upper-bound as follows

$$\mathcal{L}^* \approx F^*. \quad (16)$$

Finally, $\mathbf{c}^*, \theta_{\text{LoS}}^*, \theta_{\text{NLoS}}^*$ are considered as the user location and channel parameters estimates, respectively, for problem (6). The different steps of this algorithm is summarized in Algorithm 1.

Algorithm 1: Map-aided single user localization using PSO algorithm

- 1: Initialize $\mathcal{C}^{(0)} = \{\mathbf{c}_1^{(0)}, \dots, \mathbf{c}_C^{(0)}\}, i = 0$.
 - 2: **while** $i < I$ **do**
 - 3: **for** $\mathbf{c}_j^{(i)} \in \mathcal{C}^{(i)}$ **do**
 - 4: Classify measurements into LoS/NLoS categories using the 3D map
 - 5: Solve (12) for classified measurements
 - 6: Update (14)
 - 7: Update $\mathbf{c}_j^b, \mathbf{c}^*, \theta_{\text{LoS}}^*, \theta_{\text{NLoS}}^*$
 - 8: Update particle's velocity and position
 - 9: **end for**
 - 10: $i := i + 1$
 - 11: **end while**
-

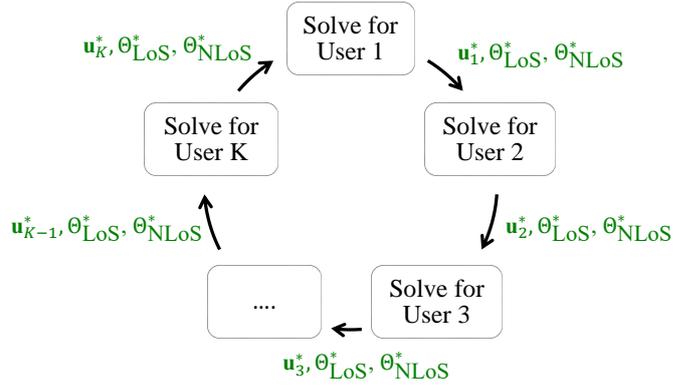


Fig. 3: Map-aided multi user localization procedure in one iteration.

C. Multi User Case

Now we proceed to the multi user case. Solving problem (6) with PSO algorithm for more than one user, even by exploiting the 3D map information is challenging, since for any possible combination of particles, an LS problem needs to be solved for finding channel parameters estimates, which is computationally complex (i.e. the complexity of the problem exponentially increases with the number of users). To tackle this difficulty, we employ a block coordinate descent technique [31] which tries to solve the original problem iteratively. Hence at each iteration, only one set of variables is updated (while fixing all the other variables), rather than updating all the variables together. More precisely, at each iteration we fix all users location estimates except one, and then we solve problem (6) for that particular user. The channel parameters are also estimated. Note that, at each iteration of the algorithm, the problem boils down to a single user case which has been addressed earlier. Doing so, the complexity of the algorithm only linearly increases with the number of users. It can be shown that, in each iteration of the proposed algorithm, the objective value defined in (5) decreases, hence the convergence is guaranteed. The proof of convergence and details of the iterative algorithm is provided in Appendix A. The procedure of multi user localization algorithm for one iteration is illustrated in Fig. 3.

IV. TRAJECTORY DESIGN FOR ACCELERATED LEARNING

When it comes to the user localization using a mobile anchor (UAV in our scenario) in a network, a natural question that arises is then as how to design its trajectory to optimally gather measurements from the users, i.e. measurements that are maximally informative about the parameters that we seek to estimate. In control and learning, such optimization framework is often termed as optimal design of experiments [32]. The relevance of this problem to our localization scenario can be understood as follows: The measurements collected from NLoS links usually lead to a degradation of the localization accuracy due to the higher shadowing effect in NLoS channels. However, designing a trajectory for the drone to establish LoS links to all users at all times is not a viable solution because there may not exist a continuous trajectory which

fulfills this constraints and because of the limited mission time. Therefore, having an autonomous trajectory design algorithm that strikes a balance between collecting LoS measurements from the users and the mission time is of the essence here.

In this section, we investigate the problem of trajectory optimization of a UAV that aims to minimize the user localization error under a limited flying time constraint. Our approach relies on the notion of Fisher information matrix (FIM) [33]. This metric helps in measuring the amount of information that observed measurements carries about an unknown user locations and channel parameters. Our goal is to exploit structural properties of the FIM, so as to design an optimal policy for the drone to collect the best possible measurements from users. Note that the works in [13], [34] have also used FIM based metric in optimizing the UAV trajectory for localization using RSS measurements, however, in those works the channel model doesn't differentiate between LoS and NLoS conditions and 3D map of the city is not exploited. In the following, we first elaborate on the FIM and its properties and then optimize the UAV's trajectory.

A. Fisher Information Matrix

For a set of channel gain measurements obtained from segment $s \in \{\text{LoS}, \text{NLoS}\}$, the FIM of the measurements with respect to the variables we seek to estimate is given by

$$\mathbf{F}_s = \mathbb{E} \left[\frac{\partial \mathcal{L}_s}{\partial \boldsymbol{\xi}_s} \frac{\partial \mathcal{L}_s}{\partial \boldsymbol{\xi}_s}^T \right], \quad (17)$$

where $\boldsymbol{\xi}_s = [\alpha_s, \beta_s, x_1, y_1, \dots, x_K, y_K]^T$, and \mathcal{L}_s is the log-likelihood of the measurements collected from segment s and is defined as

$$\mathcal{L}_s \triangleq \sum_{n=1}^N \sum_{k \in \mathcal{K}_{n,s}} \log f_{n,k,s}, \quad (18)$$

where $f_{n,k,s}$ is the probability density function (PDF) of the n -th measurement from user k which is defined in (2) conditioned on belonging to segment s , and $\mathcal{K}_{n,s}$ is a set of user indices that are in segment s at time step n . Then the FIM for all measurements in segment $s \in \{\text{LoS}, \text{NLoS}\}$ up to time step N is given by

$$\begin{aligned} \mathbf{F}_{N,s} &= \sum_{n=1}^N \sum_{k \in \mathcal{K}_{n,s}} \mathbf{H}_{n,k,s} \\ &= \mathbf{F}_{N-1,s} + \sum_{k \in \mathcal{K}_{N,s}} \mathbf{H}_{N,k,s}, \end{aligned} \quad (19)$$

where $\mathbf{H}_{n,k,s}$ is derived in Appendix B. Note that, (19) implies that the FIM is cumulative over time.

B. Cramér-Rao Bound Analysis

According to the Cramér-Rao bound (CRB) [35], the mean square error (MSE) of the estimated parameters $\hat{\boldsymbol{\xi}}_s$ for an unbiased estimator is lower bounded by

$$\text{MSE}(\hat{\boldsymbol{\xi}}_s) \geq \text{tr}(\mathbf{F}_{N,s}^{-1}). \quad (20)$$

Lemma 1. For a set of measurements collected in segment $s \in \{\text{LoS}, \text{NLoS}\}$, $\mathbf{F}_{N,s}^{-1}$ follows a recursive relation given by

$$\mathbf{F}_{N,s}^{-1} = \mathbf{F}_{1,s}^{-1} - \sum_{n=2}^N \mathbf{R}_{n,s}, \quad (21)$$

where $\mathbf{R}_{n,s}$ is defined as the amount of improvement in the estimate within time slot n .

Proof. Considering the cumulative property of $\mathbf{F}_{N,s}$, we can write

$$\begin{aligned} \mathbf{F}_{N,s}^{-1} &= \left[\mathbf{F}_{N-1,s} + \sum_{k \in \mathcal{K}_{N,s}} \mathbf{H}_{N,k,s} \right]^{-1} \\ &\stackrel{(a)}{=} \mathbf{F}_{N-1,s}^{-1} - \mathbf{R}_{N,s} \\ &\stackrel{(b)}{=} \mathbf{F}_{1,s}^{-1} - \sum_{n=2}^N \mathbf{R}_{n,s}, \end{aligned} \quad (22)$$

where (a) follows from the matrix inversion lemma, and (b) follows from the recursive relation. $\mathbf{R}_{n,s}$ is given by

$$\mathbf{R}_{n,s} = \mathbf{F}_{n-1,s}^{-1} \left(\sum_{k \in \mathcal{K}_{n,s}} \mathbf{H}_{n,k,s}^{-1} + \mathbf{F}_{n-1,s}^{-1} \right)^{-1} \mathbf{F}_{n-1,s}^{-1}. \quad (23)$$

□

C. Trajectory Optimization

We are interested to find a trajectory for the drone in limited flight mission time during which the drone starts from the base point \mathbf{v}_I and ends up at the terminal point \mathbf{v}_F while minimizes the estimation error of parameters $\boldsymbol{\xi}_s$. Such optimization problem can be formulated as

$$\min_{\{\mathbf{v}[n]\}_{n=1}^N} \text{MSE}(\hat{\boldsymbol{\xi}}_{\text{LoS}}) + \text{MSE}(\hat{\boldsymbol{\xi}}_{\text{NLoS}}) \quad (24a)$$

$$\text{s.t.} \quad (1), \mathbf{v}[1] = \mathbf{v}_I, \mathbf{v}[N] = \mathbf{v}_F, \quad (24b)$$

where $\{\mathbf{v}[n]\}_n^N = \{\mathbf{v}[n], \mathbf{v}[n+1], \dots, \mathbf{v}[N]\}$, $\text{MSE}(\hat{\boldsymbol{\xi}}_s)$ is the MSE of the estimated parameters $\hat{\boldsymbol{\xi}}_s, s \in \{\text{LoS}, \text{NLoS}\}$ by using the map-aided estimator proposed earlier in Section III-C, and constraints in (1) captures the UAV dynamics.

The optimization problem (24) is challenging to solve since a closed form expression for $\text{MSE}(\hat{\boldsymbol{\xi}}_s), s \in \{\text{LoS}, \text{NLoS}\}$ for the map-based estimator using PSO is not available. Therefore, instead of solving (24), we find a trajectory to minimize the CRB. Note that even though CRB in (20) only applies for unbiased estimators, in this case it nonetheless provides a good metric for finding favorable measurements in our trajectory optimization.

Using the CRB in (20), we can approximate (24) as

$$\min_{\{\mathbf{v}[n]\}_{n=1}^N} \text{tr}(\mathbf{F}_{N,\text{LoS}}^{-1} + \mathbf{F}_{N,\text{NLoS}}^{-1}) \quad (25a)$$

$$\text{s.t.} \quad (24b). \quad (25b)$$

Even after this approximation of the original problem, (25) is still not easy to solve. The non-linearity nature of the channel gain measurement on channel parameters and user locations makes $\mathbf{F}_{N,s}^{-1}$ a function of the unknown parameters $\boldsymbol{\xi}_s$ (that

need to be estimated), hence can not be computed exactly. To avoid this, we take a sequential approach where at time step n , we use an estimate of $\mathbf{F}_{N,s}^{-1}$, denoted by $\hat{\mathbf{F}}_{N,s|n}^{-1}$, as the objective to minimize. This estimate is obtained from using $\hat{\xi}_{s,n}$, which represents an estimate of ξ_s obtained from measurements collected up to the time slot n . So essentially, (25) becomes as an online learning and trajectory design problem in which at each time step during the mission after obtaining new measurements, unknown parameter estimates are updated, and accordingly, a new trajectory from that point is generated. The steps in the algorithm is summarized in Algorithm 2.

Algorithm 2: On-line trajectory design algorithm

- 1: Initialize $n = 1$, $\mathbf{v}[n] = \mathbf{v}_I$.
 - 2: Collect new measurements from K users
 - 3: Estimate $\hat{\xi}_{s,n}$ from measurements $[g_{1,1}, g_{1,2}, \dots, g_{n,K}]$
 - 4: Solve (26)
 - 5: UAV moves to location $\mathbf{v}^*[n+1]$
 - 6: $n := n + 1$
 - 7: Repeat Step 2, until $n < N$.
-

In Algorithm 2, the trajectory design problem in the n -th time step is given by

$$\min_{\{\mathbf{v}[n]\}_n^N} \text{tr}(\hat{\mathbf{F}}_{N,\text{LoS}|n}^{-1} + \hat{\mathbf{F}}_{N,\text{NLoS}|n}^{-1}) \quad (26a)$$

$$\text{s.t.} \quad (1), \mathbf{v}[n] = \mathbf{x}_n, \mathbf{v}[N] = \mathbf{v}_F, \quad (26b)$$

where

$$\mathbf{x}_n = \begin{cases} \mathbf{v}_I, & n = 1 \\ \mathbf{v}[n], & n > 1 \end{cases}.$$

To solve (26), by using the recursive property of $\mathbf{F}_{N,s}^{-1}$ explained in Lemma 1, dynamic programming (DP) [36] can be applied by discretizing the search space comprising discrete states (i.e. each state is a feasible discrete location in the 3D space where the drone can travel to). We denote the trajectory generated by solving (26) as $\{\mathbf{v}^*[n]\}_n^N$.

When using DP the number of computations needed to find the trajectory for Algorithm 2 equals to

$$\frac{\mathcal{V}^2 N(N-1)}{2}, \quad (27)$$

where \mathcal{V} is the number of discrete states in the search space. Since we are interested to find a 3D trajectory, \mathcal{V} is not small. To deal with this problem, in the following we elaborate on a low-complexity greedy algorithm to find a sub-optimal trajectory.

D. Greedy Trajectory Design

As mentioned in Algorithm 2, at each time step the future trajectory needs to be updated and then the drone takes the next step based on the solution obtained. Instead of designing the entire trajectory at each step, which incurs a very high complexity, we propose a greedy or myopic approach where the trajectory is designed locally and piece by piece. At each time step, the UAV seeks to find the best location for the

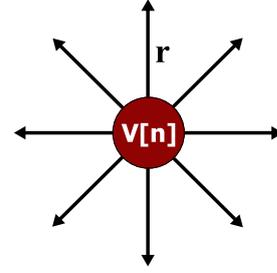


Fig. 4: An example of the possible actions in the greedy trajectory design at the n -th time step.

next step to go and collect measurements that can potentially offer the maximum improvement in the estimation error. Let us define the objective function in the greedy algorithm at time step n as

$$L(\mathbf{v}[n]) = \begin{cases} \text{tr}(\hat{\mathbf{R}}_{n,\text{LoS}|n-1} + \hat{\mathbf{R}}_{n,\text{NLoS}|n-1}), & \|\mathbf{v}[n] - \mathbf{v}_F\| \leq \tau \\ -\infty, & \text{otherwise} \end{cases}, \quad (28)$$

where $\hat{\mathbf{R}}_{n,s|n-1}$ is the estimation of $\mathbf{R}_{n,s}$ given $\hat{\xi}_{s,n-1}$, and $\tau = \rho_{max}(N-n)$. Note that the relation between $\hat{\mathbf{R}}_{n,s|n-1}$ and $\hat{\mathbf{F}}_{N,s|n-1}$ can be inferred from (22). The definition in (28) imposes the drone to reach the terminal point \mathbf{v}_F within the total flying time constraint.

Finally, the next optimal drone position at any time step $n \in [1, N-1]$ can be obtained by solving

$$\max_{\phi[n], \psi[n], \rho[n]} L(\mathbf{v}[n+1]) \quad (29)$$

$$\text{s.t.} \quad (1), \mathbf{v}[1] = \mathbf{v}_I.$$

To solve problem (29), we initialize $n = 1$ and the drone starts flying from base point $\mathbf{v}[1] = \mathbf{v}_I$. To find the best drone position in the next time step ($\mathbf{v}[n+1], n \in [1, N-1]$), we discretize the search space around the current drone location and we calculate (28) for all adjacent points. Then the neighbor point with the maximum value is chosen as the drone location in the next step. If the value of all the adjacent point are calculated as infinity, then the drone moves towards the terminal location \mathbf{v}_F by $\frac{\|\mathbf{v}[n] - \mathbf{v}_F\|}{N-n}$ meters, where n is the current time step. In Fig. 4, an example of the greedy trajectory design at the n -th time step is shown. In this example, the set of feasible drone positions comprises eight adjacent points of current drone location by selecting the input actions as follows

$$\phi[n] \in \left\{ 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4} \right\},$$

$$\psi[n] = 0, \rho[n] \in \{r\}, \quad (30)$$

where r denotes the discretization unit. To find the best position for the drone, in the next time step, all eight drone's adjacent positions at time step n need to be evaluated.

V. NUMERICAL RESULTS

In this section, we provide numerical results to show the performance of the proposed algorithms. We consider a dense urban city neighborhood of size $600 \times 600 \text{ m}^2$, comprising buildings and regular streets. The height of the buildings is Rayleigh distributed in the range of 5 to 40 m [21]. During

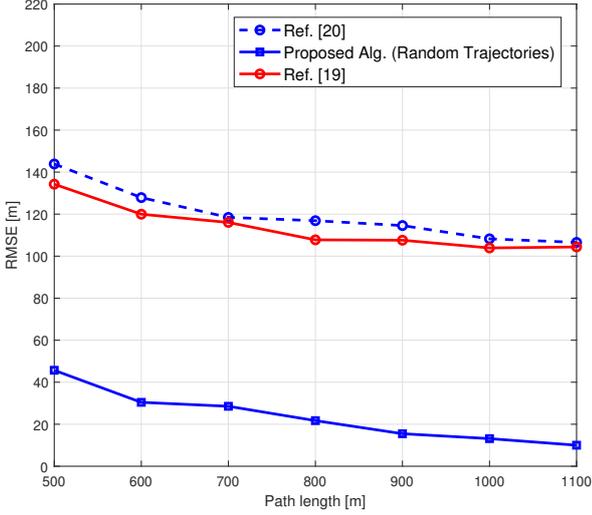


Fig. 5: Localization error performance of different algorithms (single user case) for various trajectory lengths.

the simulations, in both the cases of random and optimized trajectories, the drone starts flying from the starting point $\mathbf{v}_I = [0, 0, 50]^T$ m and ends up at the terminal points $\mathbf{v}_F = [300, 300, 50]^T$ m. The propagation parameters are chosen as $\alpha_{\text{LoS}} = 2.5$, $\alpha_{\text{NLoS}} = 3.04$, $\beta_{\text{LoS}} = -30$ dB, $\beta_{\text{NLoS}} = -35$ dB according to an urban micro scenario in [28]. The variances of the shadowing component in LoS and NLoS scenarios are $\sigma_{\text{LoS}}^2 = 5$ dB and $\sigma_{\text{NLoS}}^2 = 10$ dB, respectively.

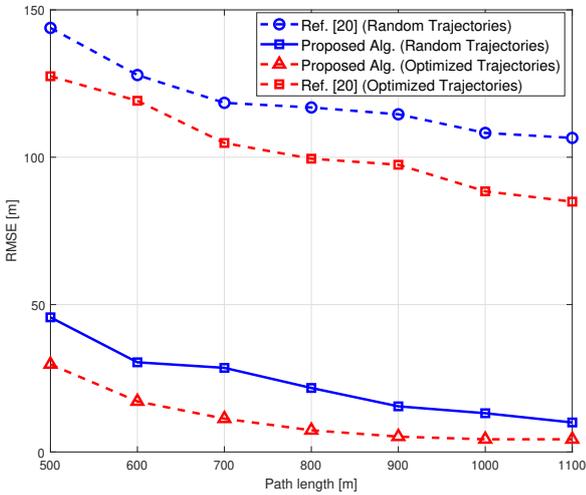


Fig. 6: Localization error (single user case) for different trajectory lengths.

In Fig. 5, we compare the performance for the localization algorithm proposed in this paper with that of [19] and [20] over random Monte-Carlo iterations in a single user case. The difference between existing methods in [19], [20], and our method is as follows: In [19] and [20], a set of arbitrary RSS measurements is used for the purpose of user local-

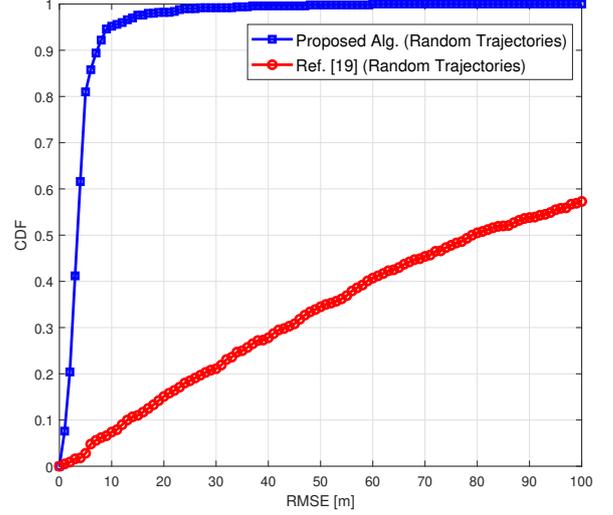


Fig. 7: CDF of single user localization accuracy.

ization and channel estimation (no trajectory optimization is considered) by assuming a two-segment radio channel model without exploiting the 3D map information. In [19], an EM algorithm is utilized for RSS measurements classification task and estimating the radio channel parameters and user locations, while in [20], an unsupervised learning method is introduced to jointly classify the RSS measurements and learn the channel model parameters and users location. The root mean square error (RMSE) of the user localization for our proposed map-based algorithm is superior to the map-unaware algorithms in [19], [20]. To have a fair comparison, a set of random trajectories was generated and then used for all algorithms. As intuited, the 3D map bring substantial improvement to the localization accuracy.

In Fig. 6, we evaluate the effect of the trajectory optimization on the localization accuracy for the map-based algorithm versus different trajectory lengths. We can see that by following the optimal trajectory, the localization error is smaller, since the UAV tries to collect the most informative measurements from the ground users.

For further comparison, we also proposed an algorithm to generate an optimized trajectory for the localization method in [20] by using the global LoS probability [37]. In this approach, we consider the same trajectory design algorithm as proposed in Algorithm 2 with the difference that instead of using the 3D map information, we assign a LoS probability for each user. Then the LoS probability of a link between k -th user and the drone at time step n is given by

$$p_k[n] = \frac{1}{1 + \exp(-a\theta_k[n] + b)}, \quad (31)$$

where $\theta_k[n] = \arctan(z[n]/r_k[n])$ denotes the elevation angle and $r_k[n]$ is the ground projected distance between the drone and the k -th node located at \mathbf{u}_k in the time slot n . Parameters $\{a, b\}$ are the model coefficients which are computed according to [37] and based on the characteristics of the city. We then assume that the k -th user is in LoS condition to the

drone at time step n if its LoS probability is greater than 0.5. The localization RMSE pertaining to this method is shown by the red dashed-line marked with squares in the figure below. It is clear that the map-based methods (both for the random and the optimized trajectories) outperform the other approaches.

In Fig. 7, the cumulative distribution function (CDF) vs. localization RMSE for different approaches in a single user scenario is shown. To localize the user using our purposed algorithm the UAV follows the optimized trajectory. Note that for all approaches, the length of the trajectory is fixed and equals to 1000 m. It can be seen that using the 3D map, the user can be localized accurately with a high probability. For example, it is almost guaranteed to localize the user with 30 meter accuracy by using the map-based approach.

In Fig. 8, we investigate the effect of increasing the number of users on the performance of the map-based localization algorithm while the drone takes a random trajectory with a fixed length of 900 m for different Monte-Carlo iterations. It is observed that the estimation error improves by increasing the number of users despite the fact that by increasing the number of users the number of unknown parameters also increases. This is because by increasing the number of users the number of gathered measurements linearly increases while the unknown parameters regarding learning the channel are fixed. Consequently, the algorithm can learn the channel more accurate which improves the localization performance as well.

An example of the optimized trajectory which is generated

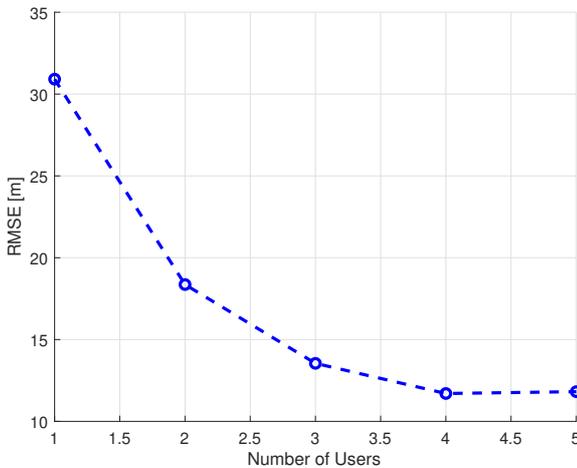


Fig. 8: Localization error when increasing the number of users.

according to Algorithm 2 for localizing 3 ground users is shown in Fig. 9. It can be seen that the users are localized accurately.

VI. CONCLUSION

In this work, we considered a UAV-aided localization system where a UAV is used as a mobile anchor to estimate the location of ground users which are randomly scattered in an urban environment. The UAV estimates the user locations from the collected RSS measurements by capitalizing on the 3D

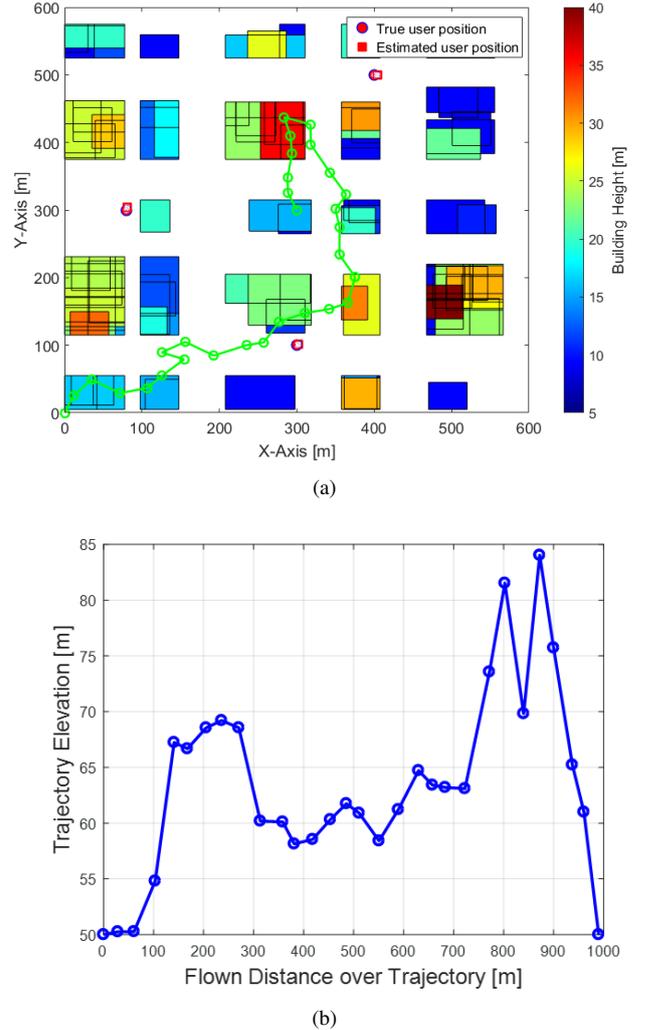


Fig. 9: (a) Top view of the generated trajectory using the sub-optimal approach. (b) Drone altitude along the trajectory.

map information of the city. Moreover, we proposed an online algorithm to design an optimized trajectory for the UAV to improve the performance of the localization under a given mission duration. The simulations show a considerable gain brought by exploiting the 3D map in the performance of the node localization compared to the conventional RSS-based localization methods. The proposed method can be advantages for the scenarios in which the GPS fails to works in the challenging environments such as dense cities where the GPS signals are obstructed by the presence of the tall buildings.

APPENDIX

A. Proof of convergence for multi-user localization

As mentioned in Section III-C, for the multi-user case we use the block coordinate descent method which is an iterative algorithm. In each iteration we fix all the user location estimates except one. Therefore, in each iteration of this algorithm the problem is recast as a single user case. For ease of exposition we merely assume one iteration for the PSO

algorithm $I = 1$. For this, to avoid notation overload, we drop the superscript indicating the PSO algorithm iteration for each particle.

Assuming we are at q -th iteration of the block coordinate descent, for the first user it can be written

$$\begin{aligned}\mathcal{L}_1^{*(q)} &:= \min_{j \in [1, C]} \mathcal{L}_1^*(\mathbf{c}_j), \\ \mathbf{u}_1^* &:= \arg \min_{\mathbf{c}_j \in C} \mathcal{L}_1^*(\mathbf{c}_j),\end{aligned}\quad (32)$$

where $\mathcal{L}_1^{*(q)}$ is the minimum cost at iteration q and is solved just for the first user, and \mathbf{u}_1^* is the corresponding user location estimate. $\mathcal{L}_1^*(\mathbf{c}_j)$ is defined as follows

$$\mathcal{L}_1^*(\mathbf{c}_j) = \min_{\boldsymbol{\theta}_{\text{LoS}}, \boldsymbol{\theta}_{\text{NLoS}}} \mathcal{L}_1(\mathbf{c}_j), \quad (33)$$

where $\mathcal{L}_1(\mathbf{c}_j)$ is the cost function for particle j by fixing all the users' location except the first user which equals to

$$\begin{aligned}\mathcal{L}_1(\mathbf{c}_j) &= \log \left(\frac{\sigma_{\text{LoS}}^2}{\sigma_{\text{NLoS}}^2} \right) |\mathcal{M}_{\text{LoS}, 1, j}| + \\ &\quad \sum_{s \in \{\text{LoS}, \text{NLoS}\}} \sum_{n \in \mathcal{M}_{s, 1, j}} \frac{1}{\sigma_s^2} |g_{n, 1} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{c}_j^{(i)})|^2 + \\ &\quad \sum_{k=2}^K \log \left(\frac{\sigma_{\text{LoS}}^2}{\sigma_{\text{NLoS}}^2} \right) |\widehat{\mathcal{M}}_{\text{LoS}, k}| + \\ &\quad \sum_{s \in \{\text{LoS}, \text{NLoS}\}} \sum_{n \in \widehat{\mathcal{M}}_{s, k}} \frac{1}{\sigma_s^2} |g_{n, k} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k^*)|^2,\end{aligned}\quad (34)$$

where \mathbf{u}_k^* is the k -th user location estimate available from the last iteration ($q - 1$), and $\widehat{\mathcal{M}}_{s, k}$ is a set of time indices of measurements collected from user k which are at segment s by assuming that user k is located at \mathbf{u}_k^* . In general, $\mathcal{L}_k(\mathbf{c}_j)$ has the form in (35).

$$\begin{aligned}\mathcal{L}_k(\mathbf{c}_j) &= \log \left(\frac{\sigma_{\text{LoS}}^2}{\sigma_{\text{NLoS}}^2} \right) |\mathcal{M}_{\text{LoS}, k, j}| + \\ &\quad \sum_{s \in \{\text{LoS}, \text{NLoS}\}} \sum_{n \in \mathcal{M}_{s, k, j}} \frac{1}{\sigma_s^2} |g_{n, k} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{c}_j^{(i)})|^2 + \\ &\quad \sum_{\substack{m=1 \\ m \neq k}}^K \log \left(\frac{\sigma_{\text{LoS}}^2}{\sigma_{\text{NLoS}}^2} \right) |\widehat{\mathcal{M}}_{\text{LoS}, m}| + \\ &\quad \sum_{s \in \{\text{LoS}, \text{NLoS}\}} \sum_{n \in \widehat{\mathcal{M}}_{s, m}} \frac{1}{\sigma_s^2} |g_{n, m} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_m^*)|^2.\end{aligned}\quad (35)$$

Now by having estimated the first user location, we then proceed to find the second user position as follows

$$\mathcal{L}_2^{*(q)} := \min_{j \in [1, C]} \mathcal{L}_2^*(\mathbf{c}_j), \quad (36)$$

and \mathbf{u}_2^* is computed similar to (32). It can be written now

$$\mathcal{L}_1^{*(q)} \geq \mathcal{L}_2^{*(q)}. \quad (37)$$

Inequality (37) holds since the PSO algorithm guarantees improvement in the cost function. In a similar manner, for all the users it can be shown that

$$\mathcal{L}_1^{*(q)} \geq \mathcal{L}_2^{*(q)} \geq \mathcal{L}_3^{*(q)} \geq \dots \geq \mathcal{L}_K^{*(q)}. \quad (38)$$

Then by proceeding to the next iteration ($q + 1$) we have

$$\mathcal{L}_1^{*(q)} \geq \mathcal{L}_1^{*(q+1)}. \quad (39)$$

And due to the fact that the MLE is lower bounded by zero then the convergence is proved.

B. Derivation of FIM

For ease of exposition, we derive FIM for the single-user case ($K = 1$). Following (2), the channel gain measurement is modeled as a Gaussian random variable with $\mathcal{N}(\lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k), \sigma_s^2)$, $s \in \{\text{LoS}, \text{NLoS}\}$. Then the PDF of each measurement equals to

$$f_{n, k, s} = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp \left(-\frac{(h_{n, k, s} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k))^2}{2\sigma_s^2} \right), \quad (40)$$

where $h_{n, k, s}$ is the random variable assigned to the measurement collected from user k at the n -th time step given segment s . Now we compute the derivative of log-likelihood as follows

$$\frac{\partial \mathcal{L}_s}{\partial \boldsymbol{\xi}_s} = \sum_{n=1}^N \sum_{k \in \mathcal{K}_{n, s}} \frac{\partial \log f_{n, k, s}}{\partial \boldsymbol{\xi}_s}, \quad (41)$$

where

$$\begin{aligned}\frac{\partial \log f_{n, k, s}}{\partial \boldsymbol{\xi}_s} &= \frac{1}{\sigma_s^2} \begin{bmatrix} -2 \log_{10} \|\mathbf{v}[n] - \mathbf{u}_k\| \\ 1 \\ \frac{-\alpha_s(x_k - x[n])}{\|\mathbf{v}[n] - \mathbf{u}_k\|^2 \log 10} \\ \frac{-\alpha_s(y_k - y[n])}{\|\mathbf{v}[n] - \mathbf{u}_k\|^2 \log 10} \end{bmatrix} \\ &\quad \times (h_{n, k, s} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k)) \\ &\triangleq \frac{(h_{n, k, s} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k))}{\sigma_s^2} \begin{bmatrix} \ell_{n, k, s}^\alpha \\ 1 \\ \ell_{n, k, s}^x \\ \ell_{n, k, s}^y \end{bmatrix}.\end{aligned}\quad (42)$$

Then the FIM is given by

$$\mathbf{F}_{N, s} = \mathbb{E} \left[\frac{\partial \mathcal{L}_s}{\partial \boldsymbol{\xi}_s} \frac{\partial \mathcal{L}_s}{\partial \boldsymbol{\xi}_s}^T \right] = \sum_{n=1}^N \sum_{k \in \mathcal{K}_{n, s}} \mathbf{H}_{n, k, s}, \quad (43)$$

where

$$\mathbf{H}_{n, k, s} = \frac{1}{\sigma_s^2} \begin{bmatrix} (\ell_{n, k, s}^\alpha)^2 & \ell_{n, k, s}^\alpha & \ell_{n, k, s}^\alpha \ell_{n, k, s}^x & \ell_{n, k, s}^\alpha \ell_{n, k, s}^y \\ (\ell_{n, k, s}^\alpha) & 1 & \ell_{n, k, s}^x & \ell_{n, k, s}^y \\ \ell_{n, k, s}^x & \ell_{n, k, s}^x & (\ell_{n, k, s}^x)^2 & \ell_{n, k, s}^x \ell_{n, k, s}^y \\ \ell_{n, k, s}^y & \ell_{n, k, s}^y & \ell_{n, k, s}^x \ell_{n, k, s}^y & (\ell_{n, k, s}^y)^2 \end{bmatrix} \quad (44)$$

Note that, to calculate (43) the following results are useful

$$\mathbb{E}[h_{n, k, s}] = \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k), \quad \mathbb{E}[(h_{n, k, s} - \lambda_n(\boldsymbol{\theta}_s, \mathbf{u}_k))^2] = \sigma_s^2. \quad (45)$$

REFERENCES

- [1] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, 2007.
- [2] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirolo, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3607–3644, 2018.
- [3] H. Huang, G. Gartner, J. M. Krisp, M. Raubal, and N. V. de Weghe, "Location based services: ongoing evolution and research agenda," *Journal of Location Based Services*, vol. 12, no. 2, pp. 63–93, 2018.

- [4] R. Zekavat and R. M. Buehrer, *Handbook of position location: Theory, practice and advances*. John Wiley & Sons, 2011, vol. 27.
- [5] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, "Survey of cellular mobile radio localization methods: From 1G to 5G," *IEEE Comm Surveys Tutorials*, vol. 20, no. 2, 2018.
- [6] R. Gangula, O. Esrafilian, D. Gesbert, C. Roux, F. Kaltenberger, and R. Knopp, "Flying rebots: First results on an autonomous UAV-based LTE relay using Open Airinterface," in *IEEE SPAWC*, 2018.
- [7] A. Chowdhery and K. Jamieson, "Aerial channel prediction and user scheduling in mobile drone hotspots," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2679–2692, 2018.
- [8] H. Sallouha, M. M. Azari, A. Chiumento, and S. Pollin, "Aerial Anchors Positioning for Reliable RSS-Based Outdoor Localization in Urban Environments," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 376–379, 2017.
- [9] R. Lima, K. Das, and D. Ghose, "Support vector regression based sensor localization using UAV," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. ACM, 2019, pp. 938–945.
- [10] O. Artemenko, A. Rubina, T. Simon, and A. Mitschele-Thiel, "Evaluation of different static trajectories for the localization of users in a mixed indoor-outdoor scenario using a real unmanned aerial vehicle," in *International Conference on Ad Hoc Networks*. Springer, 2015.
- [11] Y. Ji, C. Dong, X. Zhu, and Q. Wu, "Fair-energy trajectory planning for cooperative UAVs to locate multiple targets," in *IEEE International Conference on Communications*, 2019.
- [12] F. Koohifar, I. Guvenc, and M. L. Sichitiu, "Autonomous tracking of intermittent RF source using a UAV swarm," *IEEE Access*, vol. 6, pp. 15 884–15 897, 2018.
- [13] S. A. A. Shahidian and H. Soltanizadeh, "Autonomous trajectory control for limited number of aerial platforms in RF source localization," in *3rd International Conference on Robotics and Mechatronics (ICROM)*, 2015.
- [14] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.
- [15] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Comm Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.
- [16] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Comm Surveys Tutorials*, vol. 21, no. 3, 2019.
- [17] O. M. Cliff, R. Fitch, S. Sukkarieh, D. L. Saunders, and R. Heinsohn, "Online localization of radio-tagged wildlife with an autonomous aerial robot system," in *Robotics: Science and Systems*, 2015.
- [18] X. Li, "RSS-based location estimation with unknown pathloss model," *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, 2006.
- [19] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "Received signal strength-based joint parameter estimation algorithm for robust geolocation in LOS/NLOS environments," in *IEEE ICASSP*, May 2013.
- [20] O. Esrafilian, R. Gangula, and D. Gesbert, "UAV-relay placement with unknown user locations and channel parameters," in *52nd Asilomar Conference on Signals, Systems, and Computers*, 2018.
- [21] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *IEEE Global Communications Conference*, Dec 2014.
- [22] J. Chen, U. Yatnalli, and D. Gesbert, "Learning radio maps for UAV-aided wireless networks: A segmented regression approach," in *IEEE International Conference on Communications (ICC)*, 2017.
- [23] Q.-Y. Zhou and U. Neumann, "Complete residential urban area reconstruction from dense aerial lidar point clouds," *Graphical Models*, vol. 75, no. 3, pp. 118 – 125, 2013, computational Visual Media Conference 2012.
- [24] O. Esrafilian and D. Gesbert, "3D city map reconstruction from UAV-based radio measurements," in *IEEE Global Communication Conference (GLOBECOM)*, 2017.
- [25] Q. D. Vo and P. De, "A survey of fingerprint-based outdoor localization," *IEEE Comm Surveys Tutorials*, vol. 18, no. 1, pp. 491–506, 2016.
- [26] F. Montorsi, F. Pancaldi, and G. M. Vitetta, "Map-aware models for indoor wireless localization systems: An experimental study," *IEEE Trans on Wireless Comm*, vol. 13, no. 5, pp. 2850–2862, May 2014.
- [27] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun*, vol. 16, no. 6, pp. 3747–3760, 2017.
- [28] 3GPP TR 38.901 version 14.0.0 Release 14, "Study on channel model for frequencies from 0.5 to 100 GHz," ETSI, Tech. Rep., 2017.
- [29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *International Conference on Neural Networks (ICNN)*, 1995.
- [30] Å. Björck, *Numerical methods for least squares problems*. SIAM, 1996.
- [31] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.
- [32] L. Pronzato, "Survey paper: Optimal experimental design and some related control problems," *Automatica*, vol. 44, no. 2, p. 303–325, Feb. 2008.
- [33] S.-i. Amari and H. Nagaoka, *Methods of information geometry*. American Mathematical Soc., 2007, vol. 191.
- [34] S. A. A. Shahidian and H. Soltanizadeh, "Optimal path planning for two unmanned aerial vehicles in drss localization," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2906–2914, 2018.
- [35] A. Van den Bos, *Parameter estimation for scientists and engineers*. John Wiley & Sons, 2007.
- [36] D. E. Kirk, *Optimal control theory: an introduction*. New York, USA: Dover, 2012.
- [37] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.