IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 27, 2021, accepted June 12, 2021, date of publication June 16, 2021, date of current version June 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3089900

GTRS-Based Algorithm for UAV Navigation in Indoor Environments Employing Range Measurements and Odometry

J. P. MATOS-CARVALHO [1], RICARDO SERRAS SANTOS [1], SLAVISA TOMIC [1], AND MARKO BEKO [1,2]

[1]Cognitive and People-Centric Computing Laboratories (COPELABS), Universidade Lusófona de Humanidades e Tecnologias, 1749-024 Lisbon, Portugal [2]Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal

Corresponding author: J. P. Matos-Carvalho (joao.matos.carvalho@ulusofona.pt)

This work was supported in part by the Fundação para a Ciência e a Tecnologia under Project UIDB/04111/2020, Project foRESTER PCIF/SSI/0102/2017, and Project IF/00325/2015.

ABSTRACT This work addresses the problem of unmanned aerial vehicle (UAV) navigation in indoor environments. Due to unavailability of satellite signals, the proposed algorithm takes advantage of terrestrial radio measurements between the UAV and a set of stationary reference points, from which it extracts range information, as well as odometry by means of inertial sensors, such as accelerometer. On the one hand, based on maximum a posteriori (MAP) criterion, the range information and accumulated knowledge throughout the UAV's movement are employed to derive a generalized trust region sub-problem (GTRS), that is solved *exactly* via bisection procedure. On the other hand, by using the UAV's transform in relation to the world, another position estimation is obtained by employing odometry. Finally, the two position estimates are combined through a Kalman filter (KF) to enhance the positioning accuracy and obtain the final UAV's position estimation. The UAV is then navigated to a desired destination, by simply calculating the velocity components in the shortest path. Our results show that the proposed algorithm is robust to various model parameters for high precision (HP) UAV sensors, achieving reasonably good positioning accuracy. Besides, the results corroborate that the proposed algorithm is suitable for real-time applications, consuming (on average) only 21 ms to estimate the UAV position.

INDEX TERMS Generalized trust region sub-problem (GTRS), indoor environments, Kalman filter (KF), maximum a posteriori (MAP) estimator, navigation, odometry, positioning, unmanned aerial vehicle (UAV).

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are a recent technology that can be used in various applications [1], [2], for example, in emergency situations response [3], [4] or in warehouse inventory, and similar applications that are currently using manual methods [5]–[11]. The key challenge in most applications is to guarantee accurate navigation of the UAV [12].

UAVs navigate mainly by relying on global navigation satellite system (GNSS), such as global positioning system, that offers global coverage and work relatively well in outdoor environments. However, there are many scenarios where GNSS signal might be blocked (inside a building for example) making this technique impractical. Therefore, in situations where GNSS signal may be weak or even non-existent, UAVs require alternative sources of navigation. Due to this restraint, employing UAVs in indoor environments could represent a challenge, despite the fact that flight in such surroundings might be somewhat simpler due to absence of wind, fog, and/or rain [13], for instance.

An alternative solution could be employing a virtual sensor, comprised of an optical-flow sensor, an orientation sensor, a range sensor, and a geometric camera [14]. The data from all sensors could then be combined using a sensor fusion algorithm. The main drawback of this approach is having to install the above-mentioned sensors on the UAV, which would likely increase its mass and expenses.

Another possibility might be employing ultrasonic positioning (ULPS) combined with a time-of-flight (TOF) camera [15]. In this case, the ultrasonic emitter module is usually installed in the ceiling of the room and the acoustic signals emitted by this system are acquired by a portable receiver

The associate editor coordinating the review of this manuscript and approving it for publication was Jiankang Zhang.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

module on top of the UAV, while the TOF camera is layed on the floor pointing upwards. A computer is used to collect both acoustic signals acquired by the portable UAV receiver and the range-image generated by the TOF camera, which are used to compute the UAV's position. This approach has certain shortcomings like having to install the TOF camera, the ULPS sensor on the UAV, and the use of the computer to calculate the position of the UAV. There are other resembling sensors such as visible light communication (VLC) [16] and light detection and ranging (LiDAR) [17]–[19] that could also be employed for indoor positioning of UAVs.

Moreover, exploiting received signal strength (RSS) at the UAV together with a Q-learning algorithm could also be a valid alternative [20]. This approach requires no exact mathematical representation of the target nor mapping of the environment to position the target. The UAV starts from an initial position and measures the RSS at that position, assigning a state label to the particular RSS value. Based on the fact that no two grids have the same RSS value, each position is mapped uniquely by a single state. Then, the UAV makes a decision depending on the strategy of the algorithm in use and navigates towards a new location. The main drawback of this approach is that it assumes that the source of the radio signal and the environment conditions are stationary, which restricts its applicability.

Wireless fidelity (Wi-Fi) trilateration is another method that could be used [21]. Trilateration is a simple geometric approach in which one is interested in finding the intersection points of the circles centered around a reference point with radii equal to range observations extracted from a certain property of the received radio signal (e.g., TOF, RSS, etc.). Even though, this technique is very light in terms of computational complexity, its main downside is that it is usually based on low-noise assumption (usually required to *linearize* the measurement model), making them highly erroneous in surroundings where this assumption does not hold.

Algorithms based on time-difference of arrival (TDOA) measurements [22]–[28] can also be used for indoor autonomous flights. These algorithms are used in many areas such as target tracking [29], [30], seismic exploration [31], sensor networks [22], [32], and navigation [33]. The main disadvantage of algorithms based on TDOA is that they require additional (reference) devices than for instance RSS- or TOF-based ones, and their performance might depend on the choice of this reference sensor.

Ultra-wideband (UWB) sensors can be combined with TDOA [34] and particle filtering (PF) [35] to achieve higher precision. Nonetheless, UWB sensors might not be as affordable as some other ones, while the performance of PF depends on the number of particles used and its computational burden can rapidly become extensive in applications that demand high accuracy.

In [36], an indoor positioning system that localizes and tracks a tag that broadcasts Bluetooth low energy (BLE) beacon messages to BLE receivers was proposed. The scheme proposed in [36] first applies a Kalman filter (KF) to preprocess collected RSS information in order to smooth the fluctuated RSS data. Then, it applies a PF to approximate the unknown location of a tag (by computing probabilities of particles) and gradually reduce the location uncertainties in a Gaussian belief space, and RSS gradient model for the motion estimation.

This work proposes a new navigational algorithm, which is mainly based on range measurements extracted from terrestrial radio signals. By following Bayes' theory, we first combine maximum likelihood (ML) estimation and prior knowledge to form the problem as a maximum a posteriori (MAP) estimator. Due to its non-convexity, we apply certain approximations to the MAP estimator to formulate another one, which we effortlessly convert into a generalized trust region sub-problem (GTRS) framework. The main advantages of such an approach are that it allows us to bypass the difficulties of the MAP estimator (no additional approximations are required) and obtain an exact solution by merely a bisection procedure, which results in linear computational complexity in the number of reference points of the proposed solution. We then make use of odometry, which, by means of a Kalman filter, we integrate together with the solution obtained through the proposed GTRS estimator to further enhance its positioning accuracy. Finally, by calculating the direction from an obtained position estimate towards a desired destination, we navigate the UAV according to the shortest path criterion. Therefore, the main contributions of this work are three-fold and are summarized as follows:

- We propose a novel algorithm for UAV navigation in indoor environments, which is based on GTRS framework that allows for solving via a bisection procedure. The proposed algorithm is very efficient in terms of time consumption, and thus, suitable for real-time applications;
- We use an open-source Gazebo 3D robotics simulator environment to assess the quality of the proposed solution in near real-life conditions. This is an open platform that allows the integration of different inputs and is used to run multiple tasks, such as image processing, data relaying, remote control of a UAV (and more);
- Further enhancement of the positioning accuracy of the proposed solution through the use of a KF, which combines the estimated solution with another one obtained by means of odometry.

The remainder of this work is organized as follows. Section II presents the overall architecture developed and the system model proposed in this work. Section III formulates the range-based indoor positioning problem, while Section IV describes the proposed indoor navigation algorithm. Section V describes the simulator used within this work and how data are presented using the robot operating system (ROS) framework, and evaluates the performance of the proposed solution. Finally, the main findings and possible future research directions are summarized in Section VI.

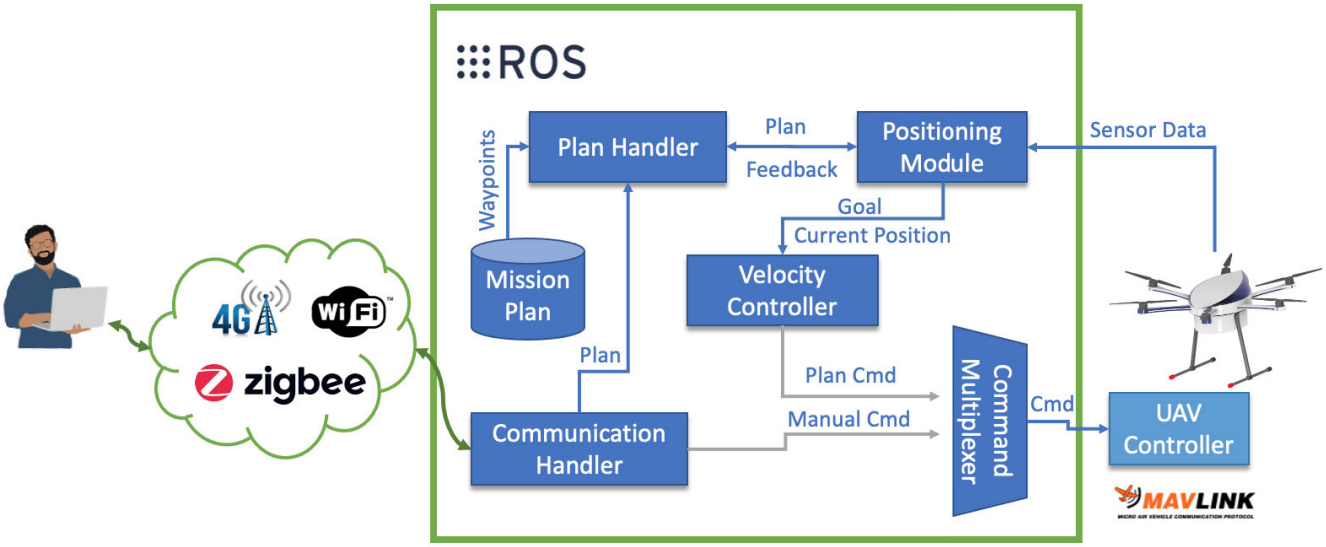


FIGURE 1. Architecture of the framework for indoor positioning with UAVs (based on [39]).

II. FRAMEWORK FOR INDOOR POSITIONING WITH AUTONOMOUS UAVs

This work proposes the architecture depicted in Figure 1. Even though the core of the work is related to positioning, the Positioning Module is also implemented within the ROS framework [37], [38]. The communication between blocks is executed throughout ROS topics and services using the publisher/subscriber paradigm. The cloud between the user and the UAV can be performed through the Wi-Fi, Zigbee and/or 4G protocols. In this way the user is able to remotely communicate with/control the UAV.

After explaining the possible communication protocols between the user and the UAV, it will be necessary to understand the five main blocks implemented according to the ROS framework (in Figure 1):

1) *Communication Handler:* The Commuication Handler block is responsible for maintaining interoperability between the user and the UAV. It is also responsible for triggering the pre-saved UAV mission through the topic ROS *localization/start* with the message type *std_msgs/EmptyMessage* [40];
2) *Plan Handler:* This block is responsible for sending each point of the total mission to the Positioning Module block through a custom service in order to increase the security of communication and the entire system pipeline [41]. In this custom service, the Positioning block asks the Plan Handler block for the next point of the mission to be reached through the variable *std_msgs/Int32 indexPathRequest*. In turn, the Plan Handler block returns the next point through the variable *geometry_msgs/Pose serverPoseResponse*, where they contain the local coordinates of the intended destination;

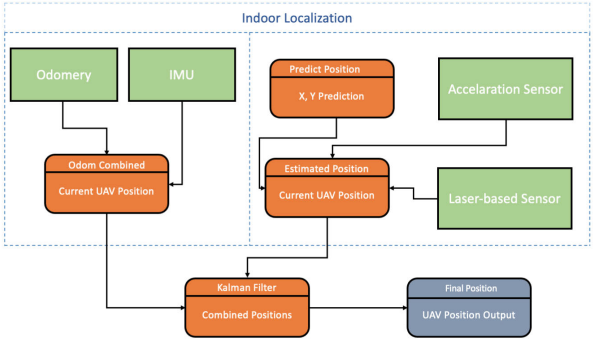


FIGURE 2. The proposed indoor positioning system model.

3) *Positioning Module:* This module is the core block of the presented architecture, as shown in Figure 1. It is here where all calculations regarding the indoor positioning of the UAV are made. Through the acceleration sensor, the laser-based sensor [42] and the algorithm proposed in Section IV, it is possible to estimate the UAV's position. Figure 2 presents more details on how this module works.
In order to enhance the positioning accuracy, a Kalman filter (KF) [43] is applied in order to merge the position estimated by the proposed algorithm and the information obtained through the use of Odometry.
The estimated position of the UAV, together with the desired destination, are then sent to the Velocity Controller block that navigates the UAV to the desired destination;
4) *Velocity Controller:* In order to navigate the UAV through the ROS mavros package [44], this block combines the two pieces received from the Positioning

Module block. With this information, the Velocity Controller block calculates the velocity required to reach the desired destination. In order to avoid dependency on proportional–integral–derivative (PID) controllers, where the variables change depending on the type of UAV, the UAV's velocity calculation on the three axes were based on [45], where:

$$\boldsymbol{eP}^{(t)} = \boldsymbol{gP}^{(t)} - \boldsymbol{cP}^{(t)} \tag{1}$$

and

$$\boldsymbol{eD}^{(t)} = ||(\boldsymbol{eP}^{(t)})|| \tag{2}$$

where $\boldsymbol{eP}^{(t)}$ represents the error position, $\boldsymbol{gP}^{(t)}$ the goal position, $\boldsymbol{cP}^{(t)}$ the current position at time instant t, and $\boldsymbol{eD}^{(t)}$ is the distance error position $\boldsymbol{eP}^{(t)}$.
With Equations 1 and 2 it is possible to normalize the error as shown in Equation 3.

$$\boldsymbol{eN}^{(t)} = \frac{\boldsymbol{eP}^{(t)}}{\boldsymbol{eD}^{(t)}} \tag{3}$$

where $\boldsymbol{eN}^{(t)}$ is the error normalized.
If the distance is lower than a certain threshold, τ (in this work, the threshold value is set to $\tau = 4$ meters), Equation 4 is activated.

$$\boldsymbol{vP}^{(t)} = \boldsymbol{eP}^{(t)} \cdot \left(\frac{\boldsymbol{eD}^{(t)}}{\tau}\right)^{SF} \tag{4}$$

where $\boldsymbol{vP}^{(t)}$ is the velocity vector and SF is the Smooth Factor (the SF was set to 2 [45]).
If the distance is higher than 4 meters (threshold), Equation 5 is then used.

$$\boldsymbol{vP}^{(t)} = \boldsymbol{eN}^{(t)} \cdot PMV \tag{5}$$

where PMV is the Param Max Velocity and is equal to 2.
In this way it is allowed to dynamically vary the UAV speed depending on the UAV distance in relation to the desired destination without any sudden changes regarding the UAV's acceleration;

5) *Command Multiplexer:* Prioritizing safety and control topics is a mandatory precaution in nowadays UAVs. Safety requires one to be able to automatically switch from autonomous behavior to manual control when pushing any button of a remote controller. Therefore, all input sources must be multiplexed into a single convergence point that communicates with the hardware controller.
The Command Multiplexer (CM) block subscribes to a list of topics, which are publishing commands and multiplexes them according to a priority criteria. The input with the highest priority controls the UAV by mavros package [44] with the mavlink protocol [46], becoming the active controller. The active controller can be changed by timeout (no response from an input) or topic locking (some inputs might be locked, being discarded). In practice, the node will take multiple input topics from different issuers and output the messages of the issuer with the highest priority (blocking the others). This is particularly useful when the UAV is flying autonomously and the pilot wants to take control of the UAV. Any command of the pilot will make him the active controller if the pilot is set with the highest priority.

III. PROBLEM FORMULATION

Let us consider a 3-dimensional wireless network comprising N fixed reference points, whose (known) true positions are denoted by $\boldsymbol{a}_i$, $i = 1, \ldots, N$, and a moving target (UAV), whose (unknown) true position at time instant t is denoted by $\boldsymbol{x}^{(t)}$. It is assumed that the target is moving according to a nearly constant velocity motion model. Therefore, its velocity elements in each direction at time t are given by

$$\boldsymbol{v}^{(t)} = \boldsymbol{v}^{(t-1)} + \boldsymbol{r}_v^{(t)}, \tag{6}$$

where $\boldsymbol{r}_v^{(t)}$ represents the noise perturbations (due to wind gusts or slight speed corrections, for instance). Hence, according to the equations of motion [47], the position of the target at time instant t is given by

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \boldsymbol{v}^{(t-1)}\Delta + \boldsymbol{r}_x^{(t)}, \tag{7}$$

with Δ and $\boldsymbol{r}_x^{(t)}$ being the sampling interval among two successive time steps and position process noise respectively. Thus, by describing the target state at time instant t by both its position and velocity, i.e., $\boldsymbol{\theta}^{(t)} = \left[\left(\boldsymbol{x}^{(t)}\right)^T, \left(\boldsymbol{v}^{(t)}\right)^T\right]^T \in \mathbb{R}^6$, from (6) and (7) it follows that

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{S}\,\boldsymbol{\theta}^{(t-1)} + \boldsymbol{r}^{(t)}, \tag{8}$$

where

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 & 0 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

denotes the state transition matrix, whereas $\boldsymbol{r}^{(t)} = [(\boldsymbol{r}_{\boldsymbol{x}}^{(t)})^T, (\boldsymbol{r}_{\boldsymbol{v}}^{(t)})^T]^T$ stands for the state process noise [48], [49], assumed to be a zero-mean Gaussian random variable with covariance matrix $\boldsymbol{Q}$, i.e., $\boldsymbol{r}^{(t)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$. The covariance matrix of the state process noise is given by

$$\boldsymbol{Q} = q \begin{bmatrix} \frac{\Delta^3}{3} & 0 & 0 & \frac{\Delta^2}{2} & 0 & 0 \\ 0 & \frac{\Delta^3}{3} & 0 & 0 & \frac{\Delta^2}{2} & 0 \\ 0 & 0 & \frac{\Delta^3}{3} & 0 & 0 & \frac{\Delta^2}{2} \\ \frac{\Delta^2}{2} & 0 & 0 & \Delta & 0 & 0 \\ 0 & \frac{\Delta^2}{2} & 0 & 0 & \Delta & 0 \\ 0 & 0 & \frac{\Delta^2}{2} & 0 & 0 & \Delta \end{bmatrix},$$

where q represents the intensity of the state process noise [49], [50].

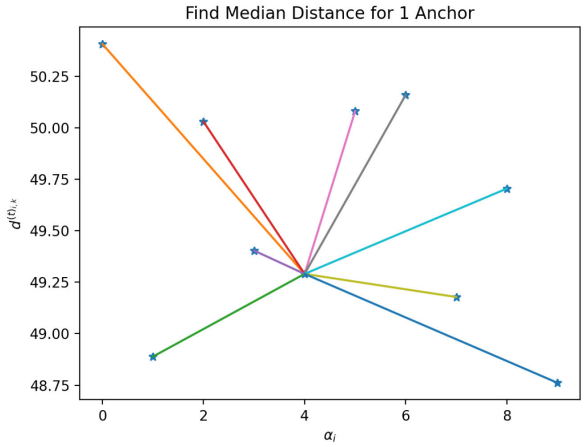


FIGURE 3. Median $d_{i,k}^{(t)}$ calculation for one a_i.

Moreover, it is assumed that the target sends a radio signal at each time instant to the reference points, which are assumed suitably equipped to withdraw the distance quantity from the received signal (e.g., from RSS, TOF, TDOA, etc.). The k-th sample of the distance between the i-th reference point and the target at time instant t is simply modeled by:

$$d_{i,k}^{(t)} = \|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\| + n_{i,k}^{(t)}, \tag{9}$$

with $\|\bullet\|$ and $n_{i,k}^{(t)}$ representing respectively the Euclidean norm and measurement noise, modeled as a zero-mean Gaussian random variable with standard deviation $\sigma_{i,k}^{(t)}$, i.e., $n_{i,k}^{(t)} \sim \mathcal{N}\left(0, \sigma_{i,k}^{(t)}\right), k = 1, \ldots, K$. It is worth mentioning that $\sigma_{i,k}^{(t)}$ is inversely proportional to signal-to-noise ratio, i.e., to the quality of the radio measurement. Although we employ the median of the K measurements as shown in Fig. 3, for the sake of notation simplicity and without loss of generality, we assume that $K = 1$ in order to remove the subscript k in the following derivations.

By following Bayesian philosophy, knowledge accumulated through the state transition model in (8) during time, combined with observations in (9), results in a marginal posterior PDF, $p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t)})$. The reason for interest in the marginal posterior is that it quantifies the belief one has in the values of the state, $\boldsymbol{\theta}^{(t)}$, given all prior observations (from all reference points), $\boldsymbol{d}^{(1:t)}$, from which an estimate at any time instant can be obtained. We sum up the key parts of the Bayesian philosophy in the following [47].

- *Initialization:* Set the marginal posterior PDF at $t = 0$ equal to the prior PDF, $p(\boldsymbol{\theta}^{(0)})$, of $\boldsymbol{\theta}^{(0)}$.
- *Prediction:* Follow the state transition model in (8) and use the measurements up to $t-1$ to obtain the predictive density of the state at t as

$$p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)}) = \int p(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t-1)}) p(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{d}^{(1:t-1)}) d\boldsymbol{\theta}^{(t-1)}. \tag{10}$$

- *Update:* Apply Bayes' rule [49], [51], to get

$$p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t)}) = \frac{p(\boldsymbol{d}^{(t)}|\boldsymbol{\theta}^{(t)}) p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)})}{p(\boldsymbol{d}^{(t)}|\boldsymbol{d}^{(1:t-1)})}, \tag{11}$$

where $p(\boldsymbol{d}^{(t)}|\boldsymbol{\theta}^{(t)})$ represents the likelihood and

$$p(\boldsymbol{d}^{(t)}|\boldsymbol{d}^{(1:t-1)}) = \int p(\boldsymbol{d}^{(t)}|\boldsymbol{\theta}^{(t)}) p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)}) d\boldsymbol{\theta}^{(t)}$$

is just a normalizing constant; it does not depend on $\boldsymbol{\theta}^{(t)}$ and is required to assure that $p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)})$ integrates to 1 [47].

According to (11) and the MAP criterion [47], an estimator of $\boldsymbol{\theta}^{(t)}$ can be obtained as

$$\begin{aligned} \hat{\boldsymbol{\theta}}^{(t)} &= \arg\max_{\boldsymbol{\theta}^{(t)}} \; p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t)}) \\ &= \arg\max_{\boldsymbol{\theta}^{(t)}} \; p(\boldsymbol{d}^{(t)}|\boldsymbol{\theta}^{(t)})\, p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)}). \end{aligned} \tag{12}$$

Note that the density in (10) cannot be calculated in analytic form in general, since the analytical solution to the posterior PDF at $t-1$ is not available, and, if the state model is not linear, the integral in (10) has no closed-form solution. Thus, some approximations are usually necessary to find $p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t)})$. In the following section, we will show how to bypass the difficulties in solving (12) directly, by converting it into GTRS, that will allow us to efficiently estimate $\hat{\boldsymbol{\theta}}^{(t)}$ by just a bisection procedure.

In this work, not only do we aspire to determine the target's position, but our goal is to navigate it towards a desired destination, $\boldsymbol{x}_{\text{destination}}$.

IV. THE PROPOSED NAVIGATION ALGORITHM

This section describes the proposed algorithm for UAV navigation. Naturally, in order to navigate the UAV, we first need to determine its position. Hence, for the sake of a better flow of presentation, we first present the algorithm to determine the position of the target, after which, we show how to navigate it to the desired position. More precisely, we divide this section into three parts: in the first part, we present the proposed GTRS algorithm for determining UAV's position by employing radio measurements only; in the second part, we show that the UAV's position estimation can be enhanced by also integrating the UAV's estimate obtained through Odometry only, via a KF; finally, the ultimate part describes and summarizes the proposed algorithm for UAV navigation.

A. DETERMINING UAV's POSITION VIA RADIO MEASUREMENTS

The problem in (12) is composed of the maximum likelihood (ML) part, which is non-convex due to the norm term in the measurement model, and the prior PDF part. Hence, tackling it directly might be very strenuous since several local optimas might be available.

Here, we take a different approach, where we start by approximating the ML part. By introducing weights, $\boldsymbol{w}^{(t)} = [w_i^{(t)}]^T$, where $w_i^{(t)} = \frac{(d_i^{(t)})^{-1}}{\sum_{i=1}^{N} (d_i^{(t)})^{-1}}$, to assign more trust to

closer links and resorting to (9), the ML part of (12) can be approximated by another non-convex estimator as

$$\underset{\boldsymbol{x}^{(t)}}{\text{minimize}} \sum_{i=1}^{N} w_i \left(\left(d_i^{(t)}\right)^2 - \|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\|^2 \right)^2, \tag{13}$$

by simply applying the weighted least squares criterion. Developing the squared-norm term allows for the problem in (13) to be rewritten in vector form as

$$\begin{aligned} &\underset{\boldsymbol{y}^{(t)}}{\text{minimize}} \ \|\boldsymbol{A}^{(t)}\boldsymbol{y}^{(t)} - \boldsymbol{b}^{(t)}\|^2 \\ &\text{subject to} \\ &\qquad (\boldsymbol{y}^{(t)})^T \boldsymbol{D}\boldsymbol{y}^{(t)} + 2\boldsymbol{f}^T\boldsymbol{y}^{(t)} = 0, \end{aligned} \tag{14}$$

where $\boldsymbol{y}^{(t)} = \left[(\boldsymbol{x}^{(t)})^T, \|\boldsymbol{x}^{(t)}\|^2 \right]^T \in \mathbb{R}^4$ and

$$\boldsymbol{A}^{(t)} = \begin{bmatrix} \vdots & \vdots \\ \sqrt{w_i^{(t)}} 2\boldsymbol{a}_i^T & -\sqrt{w_i^{(t)}} \\ \vdots & \vdots \end{bmatrix} \in \mathbb{R}^{N\times 4},$$

$$\boldsymbol{b}^{(t)} = \begin{bmatrix} \vdots \\ \sqrt{w_i^{(t)}} \left(\|\boldsymbol{a}_i\|^2 - \left(d_i^{(t)}\right)^2 \right) \\ \vdots \end{bmatrix} \in \mathbb{R}^N,$$

and

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{1\times 3} & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \quad \boldsymbol{f} = \begin{bmatrix} \boldsymbol{0}_{3\times 1} \\ -1/2 \end{bmatrix} \in \mathbb{R}^4,$$

with $\boldsymbol{I}_G$ and $\boldsymbol{0}_{p\times q}$ denoting the identity matrix of size G and the matrix of all-zero entries of size $p \times q$. The above optimization problem consists of a quadratic objective function and a quadratic constraint; this class of optimization problems is known in the literature as a GTRS [52], [53], and, even though non-linear in general, it is possible to calculate an interval within which it is a monotonically decreasing function. Thus, GTRS is suitable for solving *exactly* by means of bisection procedure [54].

With (14), we are able to determine the UAV's position at any time instant, by resorting merely to radio observations. This estimation can, however, be significantly improved by combining the measurements with the accumulated knowledge from the prior estimations, as described in Section III. Nonetheless, to do so, certain approximations are required regarding the posterior distribution.

The posterior distribution $p(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{d}^{(1:t-1)})$ can be approximated by a Gaussian one,[1] i.e., $p(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{d}^{(1:t-1)}) \sim \mathcal{N}\left(\hat{\boldsymbol{\theta}}^{(t-1|t-1)}, \hat{\boldsymbol{P}}^{(t-1|t-1)}\right)$, the mean and the covariance matrix of which can be obtained by resorting to GTRS. Then, according to (10), we have that

$$\begin{aligned} p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)}) &\approx c \\ &\times \exp\left\{ \left(\boldsymbol{\theta}^{(t)} - \hat{\boldsymbol{\theta}}^{(t|t-1)}\right)^T \left(\hat{\boldsymbol{P}}^{(t|t-1)}\right)^{-1} \left(\boldsymbol{\theta}^{(t)} - \hat{\boldsymbol{\theta}}^{(t|t-1)}\right) \right\}, \end{aligned} \tag{15}$$

where c is a normalizing constant, and $\hat{\boldsymbol{\theta}}^{(t|t-1)}$ and $\hat{\boldsymbol{P}}^{(t|t-1)}$ are respectively the mean and the covariance of the one-step predicted state, calculated as

$$\hat{\boldsymbol{\theta}}^{(t|t-1)} = \boldsymbol{S}\hat{\boldsymbol{\theta}}^{(t-1|t-1)}, \tag{16a}$$

$$\hat{\boldsymbol{P}}^{(t|t-1)} = \boldsymbol{S}\hat{\boldsymbol{P}}^{(t-1|t-1)}\boldsymbol{S}^T + \boldsymbol{Q}. \tag{16b}$$

By using (14) and (15), we can then approximate (12) as

$$\begin{aligned} &\underset{\boldsymbol{z}^{(t)}}{\text{minimize}} \ \|\widetilde{\boldsymbol{A}}^{(t)}\boldsymbol{z}^{(t)} - \widetilde{\boldsymbol{b}}^{(t)}\|^2 \\ &\text{subject to} \\ &\qquad (\boldsymbol{z}^{(t)})^T \widetilde{\boldsymbol{D}}\boldsymbol{z}^{(t)} + 2\widetilde{\boldsymbol{f}}^T\boldsymbol{z}^{(t)} = 0, \end{aligned} \tag{17}$$

where $\boldsymbol{z}^{(t)} = \left[(\boldsymbol{\theta}^{(t)})^T, \|\boldsymbol{x}^{(t)}\|^2 \right]^T \in \mathbb{R}^7$,

$$\widetilde{\boldsymbol{A}}^{(t)} = \begin{bmatrix} \left[\sqrt{w_1^{(t)}} 2\boldsymbol{a}_1^T, \ \boldsymbol{0}_{1\times 2} \right] & -\sqrt{w_1^{(t)}} \\ \vdots & \vdots \\ \left[\sqrt{w_N^{(t)}} 2\boldsymbol{a}_N^T, \ \boldsymbol{0}_{1\times 2} \right] & -\sqrt{w_N^{(t)}} \\ \left(\hat{\boldsymbol{P}}^{(t|t-1)}\right)^{-1/2} & \boldsymbol{0}_{4\times 1} \end{bmatrix} \in \mathbb{R}^{(N+6)\times 7},$$

$$\widetilde{\boldsymbol{b}}^{(t)} = \begin{bmatrix} \sqrt{w_1^{(t)}} \left(\|\boldsymbol{a}_1\|^2 - \left(d_1^{(t)}\right)^2 \right) \\ \vdots \\ \sqrt{w_N^{(t)}} \left(\|\boldsymbol{a}_N\|^2 - \left(d_N^{(t)}\right)^2 \right) \\ \left(\hat{\boldsymbol{P}}^{(t|t-1)}\right)^{-1/2} \hat{\boldsymbol{\theta}}^{(t|t-1)} \end{bmatrix} \in \mathbb{R}^{N+6},$$

and

$$\widetilde{\boldsymbol{D}} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 4} \\ \boldsymbol{0}_{4\times 3} & \boldsymbol{0}_{4\times 4} \end{bmatrix} \in \mathbb{R}^{7\times 7}, \quad \widetilde{\boldsymbol{f}} = \begin{bmatrix} \boldsymbol{0}_{6\times 1} \\ -1/2 \end{bmatrix} \in \mathbb{R}^7.$$

The problem in (17) is a GTRS, and similarly as (14), it can be solved *exactly* via a bisection method.

B. ENHANCING POSITIONING ACCURACY VIA ODOMETRY

Another way to obtain an estimate of the UAV's position is through odometry. Through the ROS framework it is possible to get a position estimation by using the UAV transform in relation to the world using the UAV sensors, such as accelerometer, represented by (18).

$$\hat{\boldsymbol{x}}_{\text{Odom}}^{(t)} = h(\boldsymbol{acc}^{(t)}, \boldsymbol{gy}^{(t)}) \tag{18}$$

where $\boldsymbol{acc}^{(t)}$ is the 3-axis accelerometer, $\boldsymbol{gy}^{(t)}$ is the 3-axis gyroscope and h represents the transform [55] in order to discover the local position of the UAV (in this work, it is known that the $\hat{\boldsymbol{x}}_{\text{Odom}}^{(t)}$ is equal to 0 when the positioning algorithm is launched).

[1] Naturally, it should be noted that these approximations introduce certain errors, which are particularly important in the case when the posterior density cannot be well-approximated by a Gaussian distribution (for instance, when the posterior density is multi-modal).

It is worth mentioning that relying solely on odometry might become precarious in practice, since any external force (for example, wind gusts) acting on the UAV is likely to result in erroneous position estimation. Nevertheless, this information can be exploited in order complement and further enhance the accuracy of the proposed GTRS algorithm, as it is shown in the following.

In order to take advantage of these two pieces of information and enhance the positioning accuracy of the proposed algorithm, a Kalman filter [56] is applied to determine the final UAV's position at time instant t as

$$\hat{\boldsymbol{x}}^{(t)} = KF.predict(\hat{\boldsymbol{x}}^{(t)}_{\text{GTRS}},\ \hat{\boldsymbol{x}}^{(t)}_{\text{Odom}}) \tag{19}$$

C. NAVIGATING UAV TOWARDS A DESIRED DESTINATION

Having the UAV's position estimate, $\hat{\boldsymbol{x}}^{(t)}$, and the desired destination, $\boldsymbol{x}_{\text{destination}}$, at hand, the task of navigation becomes practically trivial. To navigate the UAV, we start by calculating the estimated direction in which we would like it to navigate to as

$$\phi^{(t)} = \arctan\left(\frac{x_{\text{destination,y}} - \hat{x}^{(t)}_{\text{y}}}{x_{\text{destination,x}} - \hat{x}^{(t)}_{\text{x}}}\right), \tag{20a}$$

$$\alpha^{(t)} = \arccos\left(\frac{x_{\text{destination,z}} - \hat{x}^{(t)}_{\text{z}}}{\|\boldsymbol{x}_{\text{destination}} - \hat{\boldsymbol{x}}^{(t)}\|}\right), \tag{20b}$$

where $\phi^{(t)}$ and $\alpha^{(t)}$ are the azimuth and the elevation angles at time instant t from the estimated position, $\hat{\boldsymbol{x}}^{(t)}$, with g_s denoting the s-th coordinate of the vector $\boldsymbol{g}$.

Hence, the succeeding position of the UAV is obtained according to

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{S}\left[(\boldsymbol{x}^{(t)})^T, (\boldsymbol{u}^{(t)})^T\right]^T, \tag{21}$$

with

$$\boldsymbol{u}^{(t)} = \boldsymbol{vP}^{(\boldsymbol{t})} \cdot \begin{bmatrix} \cos(\phi^{(t)})\sin(\alpha^{(t)}) \\ \sin(\phi^{(t)})\sin(\alpha^{(t)}) \\ \cos(\alpha^{(t)}) \end{bmatrix} \in \mathbb{R}^3$$

where the second factor represents the unit vector (with $\boldsymbol{vP}^{(\boldsymbol{t})} = (4) \vee (5)$).

To conclude this section, we summarize the proposed algorithm for navigation in Algorithm 1.

Algorithm 1 The Proposed GOK-JRSM Algorithm

Require: $\boldsymbol{a}_i, d_i^{(t)}, i = 1, \ldots, N, \Delta, q, \boldsymbol{S}$

//Find initial UAV location

$\hat{\boldsymbol{x}}^{(0|0)} \leftarrow (14)$

//Initialization

$\hat{\boldsymbol{\theta}}^{(0|0)} \leftarrow \left[(\hat{\boldsymbol{x}}^{(0|0)})^T, 0, 0\right]^T$

$\hat{\boldsymbol{P}}^{(0|0)} \leftarrow \boldsymbol{I}_4$

$t \leftarrow 1$

while $\|\boldsymbol{x}_{\text{destination}} - \hat{\boldsymbol{x}}^{(t)}\| \leq \tau$ **do**

//Prediction

$\hat{\boldsymbol{\theta}}^{(t|t-1)} \leftarrow (16a)$

$\hat{\boldsymbol{P}}^{(t|t-1)} \leftarrow (16b)$

//Update

$\hat{\boldsymbol{\theta}}^{(t|t)} \leftarrow (17)$

$\hat{\boldsymbol{P}}^{(t|t)} \leftarrow \left(\hat{\boldsymbol{\theta}}^{(t|t)} - \hat{\boldsymbol{\theta}}^{(t-1|t-1)}\right)\left(\hat{\boldsymbol{\theta}}^{(t|t)} - \hat{\boldsymbol{\theta}}^{(t-1|t-1)}\right)^T$

//Position estimation via GTRS

$\hat{\boldsymbol{\theta}}^{(t|t)}_{1:3}$

//Position estimation via Odometry

$\hat{\boldsymbol{x}}^{(t)}_{\text{Odom}} \leftarrow (18)$

//Final position estimation

$\hat{\boldsymbol{x}}^{(t)} \leftarrow (19)$

//Navigation

$\boldsymbol{x}^{(t+1)} \leftarrow (21)$

$t \leftarrow t + 1$

end while

V. PERFORMANCE RESULTS

This section validates the performance of the proposed algorithm through numerical results. It is organized in two parts, where the first one describes the considered simulation environment and the second one presents a set of simulation results to assess the performance of the proposed algorithm from both accuracy and operational time point of views.

A. SIMULATION ENVIRONMENT

In order to simulate the algorithms proposed in this article, the open-source Gazebo 3D robotics simulator was chosen due to three important factors [57]–[59]:

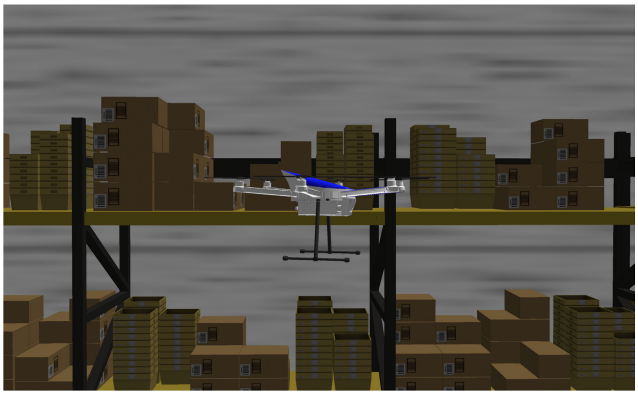
1) Its realistic environment renderings;
2) Its high-performance physics engine with a high accuracy and a wide variety of sensors like cameras, GNSS sensors, IMUs, and LIDARs;
3) The Gazebo has ROS plugins in order to be able to communicate with all simulated models.

In this way, the UAV model and the simulation environment, represented as a warehouse, were designed in Gazebo simulator as can be seen in Fig. 4. The dimensions of the simulated warehouse are set as: length: 200 meters, width: 100 meters, and height: 20 meters.

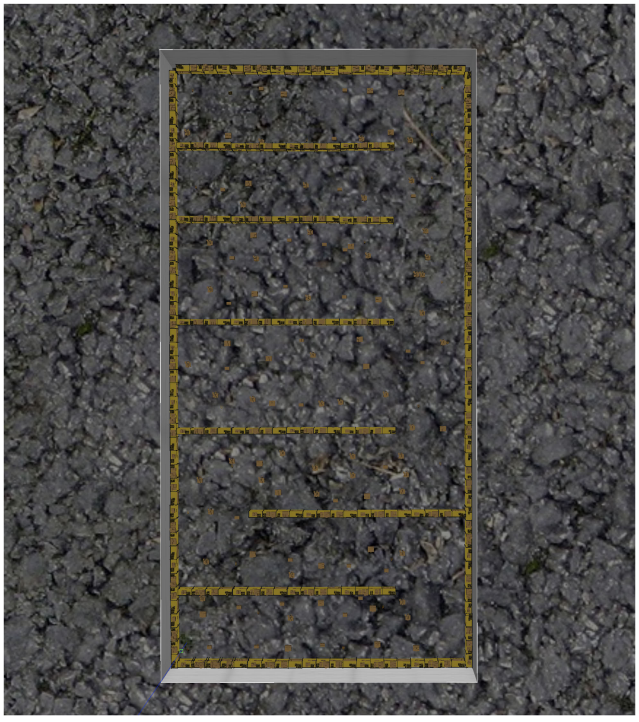
In order to be able to view all simulated UAV information by topics, services, actions (for example, acceleration sensors, odometry) in real time, a 3D visualization tool for ROS, known as RViz, was used [60].

After understanding the 3D Gazebo simulator and the RViz tool to visualize the data, the indoor navigation algorithm can be performed.

With a given set of mission points that will be sent to the UAV, represented by arrows in Fig. 5, three results are obtained for $\sigma_i^{(t)} = 1$ meter, $\forall i, \forall t$, shown in Fig. 6. The results denoted by the blue solid line is used to represent the ground truth position of the UAV, whereas the red solid line and the green solid line represent respectively the prediction of the navigation algorithm throughout the mission and the



(a)



(b)

FIGURE 4. Gazebo Simulator: (a) Unmanned Aerial Vehicles (UAV) model; and (b) warehouse model - Top view.

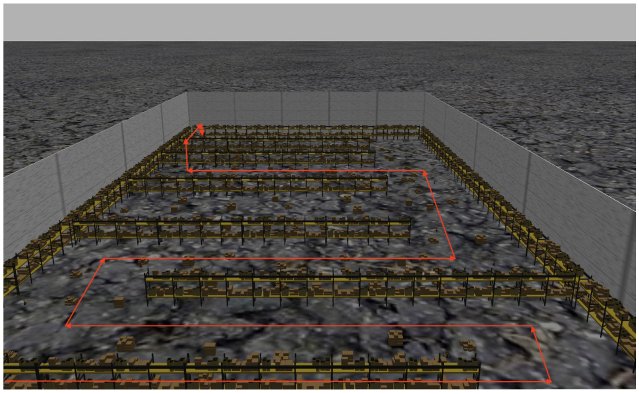
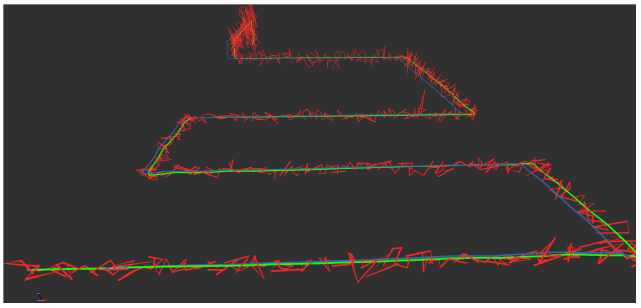
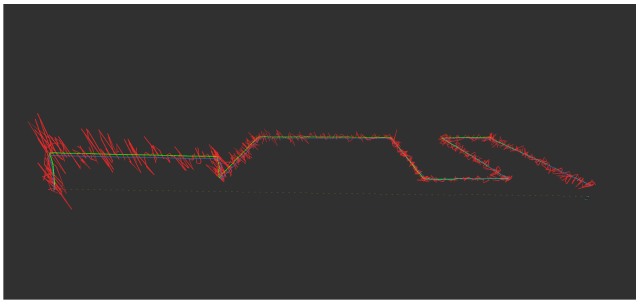


FIGURE 5. Gazebo simulator with the mission's waypoints to be executed.

combination of Odometry with the response of the navigation algorithm using KF.



(a)



(b)

FIGURE 6. RViz visualization for mission performed with $\sigma_i^{(t)} = 1$ meter, $\forall i, \forall t$: (a) Front view; and (b) side view.

From Fig. 6, it can be observed that the proposed algorithm performs well in the considered setting, given that its performance matches closely the path of the ground truth. Even so, it is necessary to highlight that in certain areas (such as when abrupt turns are made by the UAV), the gap between the true and the estimated paths are observed, as expected. This is mainly due to the fact that the accumulated prior knowledge somewhat pulls the estimate towards a new expected position (based on the prior knowledge). Nevertheless, it can be seen that the proposed algorithms recovers quickly from these imperfections, making them insignificant in comparison with the whole path.

In order to get a better comprehension of the performance of the proposed algorithm, a set of simulations is performed in order to make the conditions as close as possible to practical environments, and we present the results in Section V-B, based on Monte Carlo (Mc) criterion.

B. NUMERICAL RESULTS

Extensive simulations have been carried out in order to study the robustness of the proposed algorithm on a laptop with NVIDIA GeForce GTX 1060, 16GB RAM and Intel Core i7-9750H. In this way, it is necessary to take into account certain parameters that were used in order to evaluate this process. The number of Mission waypoints (Mw) is $Mw = 52$, the number of Monte Carlo (Mc) runs is $Mc = 100$, and, at most $N = 8$ reference points are employed whose distribution is shown in Table 1, where $B = 100$ meters with the goal that all reference points remain inside the warehouse. It is worth mentioning that first N reference points from Table 1 are always employed in all simulations presented here.

TABLE 1. True locations of the reference points in the simulation environment.

i	1	2	3	4	5	6	7	8
$\boldsymbol{a}_i$	$\begin{matrix}0\\0\\B/2\end{matrix}$	$\begin{matrix}0\\B\\B/2\end{matrix}$	$\begin{matrix}B\\0\\B/2\end{matrix}$	$\begin{matrix}B\\B\\B/2\end{matrix}$	$\begin{matrix}0\\B/5\\B/2\end{matrix}$	$\begin{matrix}0\\B/6\\B/2\end{matrix}$	$\begin{matrix}B/4\\0\\B/2\end{matrix}$	$\begin{matrix}0\\B/2\\B/2\end{matrix}$

FIGURE 7. Simulation results for three types of positioning: GOK-JRSM, Odom and Kalman Filter for $N = 4$, $K = 10$, $\sigma^{(t)} = 1$ meters $\forall i, \forall t$.

The main performance metric is the root mean square error (RMSE) [61], which is defined as

$$RMSE = \sqrt{\sum_{i=1}^{Mc} \frac{\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2}{Mc}}$$

where $\hat{\boldsymbol{x}}_i$ represents the estimate of the true source location, i.e., $\boldsymbol{x}_i$, in the i-th Monte Carlo run for a specific noise realization. An RMSE is also calculated for each intermediate mission point in order to know the impact of each block of the system at every time instant. In addition, the average RMSE (ARMSE) defined as

$$ARMSE = \sqrt{\sum_{j=1}^{Mw}\sum_{i=1}^{Mc} \frac{\|\boldsymbol{x}_{ij} - \hat{\boldsymbol{x}}_{ij}\|^2}{Mw\,Mc}},$$

is also employed as a performance metric in order to get an insight on the overall performance of the algorithm.

Fig. 7 shows the RMSE versus mission waypoints performance for $\sigma_i^{(t)} = 1$ meters, $\forall i, \forall t$. From Fig. 7 it is possible to conclude that the simulated Odometry has a somewhat higher ARMSE than the proposed positioning algorithm relying on radio measurements, but shows a more stable performance throughout the trajectory. This can be explained to some extent by the fact that the information obtain through Odometry within the simulator might represented an oversimplified view of the realistic conditions, which is the most likely reason for the observed behavior. It is questionable whether this kind of result could be obtained in real-life scenarios. Nevertheless, it is also important to note that the combination of the two positioning schemes improves the overall

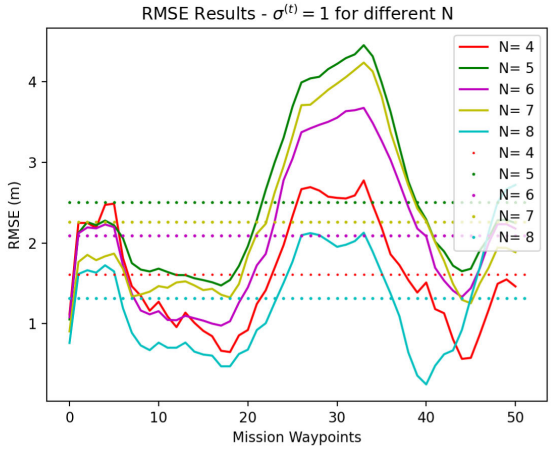


FIGURE 8. Simulation results for the proposed GTRS based algorithm with $\sigma^{(t)} = 1$ meters $\forall i$, $\forall t$ and N values between 4 and 8, with increments of 1.

position output (observed by the Kalman filter's RMSE and ARMSE calculations), obtaining a final error of 1.17 meters of ARMSE, between the Kalman Filter output and the ground truth positioning, in a trajectory length of 481 meters.

Since the main contribution of this work is related to GTRS, the robustness of this algorithm was studied in more detail. In this way, 4 processes were studied:

1) Number of reference points, N, variation;
2) Different noise power, $\sigma_i^{(t)}$, values;
3) The number of measurement samples, K, impact and the median itself for calculating the distances from the anchors to the UAV;
4) GTRS-based algorithm behaviour with and without the presence of UAV sensors.

Starting with the number of reference points, the influence of the N on the GTRS-based algorithm was studied in Fig. 8. It should be noted that in this figure, the value of $\sigma^{(t)} = 1$ meters $\forall i, \forall t$ was maintained constant in all simulations and only the number of reference points was varied from 4 to 8.

From Fig. 8 it is possible to conclude two aspects:

- As expected, the overall trend of the proposed algorithm is to improve its accuracy with the increase of N, in general. This behaviour is natural, since the greater number of reference points allows for a higher quantity of radio measurements in the network, and thus, higher positioning accuracy can be achieved;
- The positioning error is somewhat correlated with the symmetry of reference points employed in the study environment (please refer to Table 1 for the network topology), since the figure reveals better performance of the GTRS algorithms for the setting where $N = 4$ and $N = 8$, in comparison with those where $N = 5, 6$ or 7.

Studying the impact that the $\sigma_i^{(t)}$ value has on the GTRS-based algorithm, 3 different values of $\sigma_i^{(t)}$ were used for the same N, as it can be seen in Fig. 9. The figure confirms that the higher the noise power is, the worse positioning accuracy is obtained, as foreseen. Nonetheless, it is also

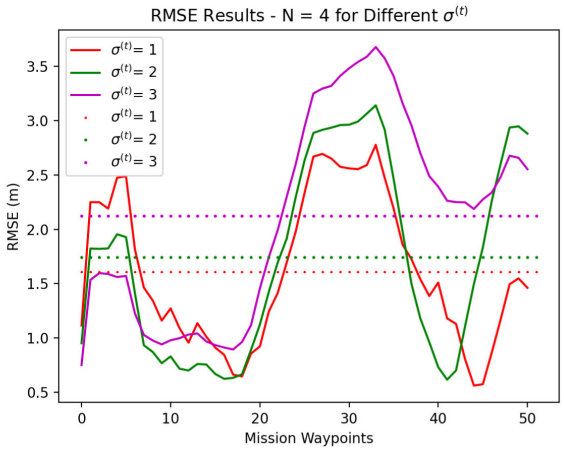


FIGURE 9. Simulation results for the proposed GTRS based algorithm with N = 4 and $\sigma^{(t)} = 1$, 2 *and* 3 meters.

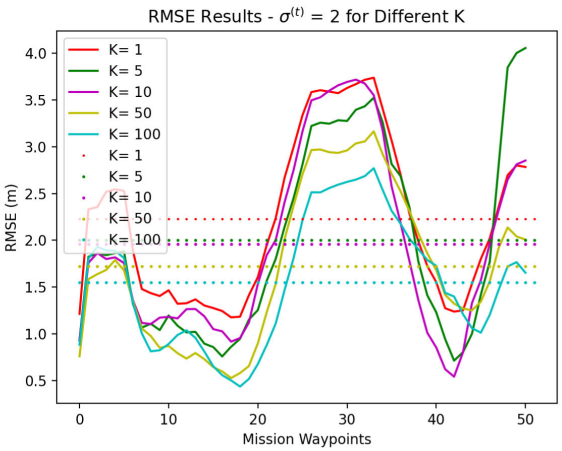


FIGURE 10. Simulation results for the proposed GTRS based algorithm with $\sigma^{(t)} = 2$ meters $\forall i, \forall t$ and K values equal to 1, 5, 10, 50 and 100.

possible to conclude that the algorithm, even with $\sigma_i^{(t)}$ equals to 1 and 2 meters, still presents an ARMSE error of less than 2 meters, while for $\sigma_i^{(t)} = 3$ meters, the ARMSE performance is somewhat above 2 meters.

Moving on to the third process studied, it is important to know what will be the impact of different K values on the GTRS based algorithm. Thus, simulations were carried out for different K values with a $\sigma_i^{(t)} = 2$ meters $\forall i, \forall t$ (to further emphasize this effect), as can be seen in Figure 10. From the figure, it is possible to conclude that the higher the value of K is, the lower the positioning error is. This is due to the simple fact that the distance will be the result of a 3-D median. With this value, the central value of the noise is chosen. This becomes an advantage since the probability of equating a high $\sigma_i^{(t)}$ value is less when compared to an average of K values. A clear disadvantage is that the noise value may also never be the lowest value obtained, which could decrease the performance of the UAV position estimation. Naturally, the large values of K (e.g., $K = 50$ or 100 leading to ARMSE = 1.72 and 1.55 meters, respectively) are preferred than lower ones (e.g., $K = 1$ or 5, leading to ARMSE = 2.23 and 2.01 meters, respectively) for $\sigma_i^{(t)} = 2$ meters.

Finally, it was necessary to know what impact the UAV sensors have on the algorithm proposed in this article. In this way, two types of distinct tests were performed: the first test was made for a case of High Precision (HP) sensors and the ARMSE was measured; the second test, the ARMSE values of the positioning algorithm were measured in case of Low Precision (LP) UAV sensors (accelerometer and gyroscope sensors have an important impact on UAV velocity output). Table 2 shows the results obtained.

From Table 2 it is possible to conclude that the algorithm based on GTRS is robust when the UAV has HP sensors. Regardless of the noise power ($\sigma_i^{(t)} = 1, 2$ and 3), the worst ARMSE result in the case of HP sensors is 0.179 meters while in the case the UAV has LP sensors the error increases for the worst case of 2.505 meters.

TABLE 2. Simulation results - HP sensors vs LP sensors.

Parameters		ARMSE (m)	
		HP Sensors	LP Sensors in [57]
$\sigma_i^{(t)} = 1$	$N = 4$	0.1793	1.6078
	$N = 5$	0.1166	2.5047
	$N = 6$	0.1033	2.0907
	$N = 7$	0.0972	2.2580
	$N = 8$	0.0809	1.3133
$N = 4$	$\sigma_i^{(t)} = 1$	0.1506	1.6078
	$\sigma_i^{(t)} = 2$	0.3831	1.7418
	$\sigma_i^{(t)} = 3$	0.8635	2.1228
$N = 4, \sigma_i^{(t)} = 2$	$K = 1$	1.0581	2.2312
	$K = 5$	0.4842	2.0043
	$K = 10$	0.4111	1.9580
	$K = 50$	0.1955	1.7225
	$K = 100$	0.1191	1.5518

Regarding the influence of $\sigma_i^{(t)}$, it is anticipated that the positioning error increases with the increase of $\sigma_i^{(t)}$. However, when the sensors are from HP, although the growth in the positioning error is significant, it actually does not reach 1 meter of ARMSE, while in the LP sensors the error exceeds 2 meters of ARMSE.

Fourth, even with the decrease in positioning error in the 2 cases from the increase in the K value, it turns out that in the worst case ($K = 1$) the positioning resolution of HP sensors is 1 meter, while, for the same conditions of K, $\sigma_i^{(t)}$ and N, the error is doubled and it decrease much more slowly than when the HP sensors are available.

TABLE 3. Processing time at each stage.

	Velocity	Odometry	GOK-JRSM
Time (ms)	256.76	256.66	**21.33**

Thus, it is concluded that the proposed GTRS-based algorithm is quite robust when the UAV sensors have a good measurement resolution. If the sensors are of LP the precision error can vary between 1 to 2 meters, which seems reasonable in several practical cases.

Finally, it is extremely important to validate whether or not an algorithm can be applied in real-time applications. Hence, we also study the running time of the proposed algorithm. In this way, the processing time of the algorithm based on the GTRS was measured and in each position estimation it took about 21 ms on average which is equivalent to approximately 47 fps; this study is a strong indicator that the proposed solution is suitable for real-time applications in terms of its running time. Table 3 shows the details of the processing time in several stages of the proposed system.

It is possible to conclude in Table 3 that the phase that uses less processing time is in the GOK-JRSM stage. It is also important to emphasize that the estimation error also depends on the time of the information coming from the sensors, as is the case of the Velocity (256.76ms). As the execution time of the GOK-JRSM algorithm is less than the Velocity processing time, the velocity estimation will have an increased error because it is not updated faster comparing with GOK-JRSM algorithm (21ms).

In addition to studying the performance of the proposed algorithm in various scenarios, it is very important to validate its performance against the existing methods as well. Here, considering its close relationship with the proposed approach and very recent date of publication, the PF method presented in [36] is used as the state-of-the-art (reference) approach. It is worth mentioning that, in all simulations presented here, the PF method is implemented with 1000 particles. Fig. 11 below illustrates the ARMSE (m) versus the number of measurement samples, K, comparison between the proposed work and PF in [36], when $\sigma = 2$m. As foreseen, both approaches improve their localization accuracy with the increase of K. The figure also exhibits that the proposed algorithm evidently outperforms the PF one for a wide span of K ($K \geq 5$), which can be explained to some extent by the fact that the new solution relies more on the quality of the measurements, whereas PF gives more trust to the measurement model and not the measurements themselves.

Table 4 below summarizes the performance comparison between the PF in [36] and the proposed solution, both in terms of localization accuracy and time consumption. The table generally shows superior localization accuracy of the proposed solution over the PF in various settings. It is also possible to observe the difference between the PF algorithm and the proposed system regarding to their time consumption: The proposed algorithm, running at 21.33ms, is roughly 10× faster than PF algorithm. These results clearly justify the use of GTRS method for the considered problem, which is possible to run it in real time.

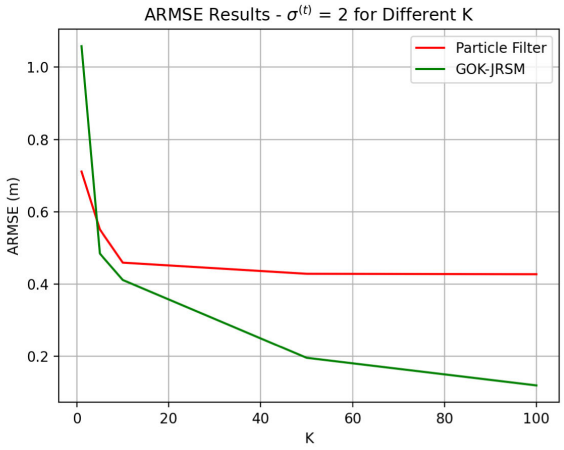


FIGURE 11. ARMSE (m) versus K.

TABLE 4. ARMSE (m) performance for different set of parameters and time consumption of PF and the proposed work.

Parameters		ARMSE (m)	
		Particle Filters in [36]	**Proposed Work**
$\sigma_i^{(t)} = 1$	$N = 4$	0.453	0.179
	$N = 5$	0.569	0.117
	$N = 6$	0.486	0.103
	$N = 7$	0.438	0.097
	$N = 8$	0.380	0.081
$N = 4$	$\sigma_i^{(t)} = 1$	0.453	0.151
	$\sigma_i^{(t)} = 2$	0.459	0.383
	$\sigma_i^{(t)} = 3$	0.513	0.864
$N = 4, \sigma_i^{(t)} = 2$	$K = 1$	0.711	1.058
	$K = 5$	0.551	0.484
	$K = 10$	0.459	0.411
	$K = 50$	0.428	0.196
	$K = 100$	0.427	0.119
Time consumption with $\sigma_i^{(t)} = 2$ $N = 6$ and $K = 100$		204.46ms	21.33ms

VI. CONCLUSION AND FUTURE WORK

This work considered the problem of UAV navigation in GNSS-less environments, where terrestrial radio signals were employed, together with Odometry. By following the MAP principle, we managed to apply favorable approximations to

the MAP estimator in order to effortlessly convert it into a GTRS framework. Its accuracy was further enhanced by integrating the positioning information acquired through Odometry, by means of a KF. After having the position estimation at hand at any time instant, the UAV's navigation was performed by simply calculating the shortest direction towards a desired destination. The simulation results corroborate the effectiveness of the proposed algorithm, both in terms of positioning accuracy and execution time, allowing it to function in real-time (21ms to estimate the UAV position).

It was also concluded in this work that the positioning algorithm is robust when the UAV sensors have a good measurement resolution, obtaining an ARMSE error that does not exceed 1 meter. However, when the UAV measurement sensors do not have a good precision, e.g. Low precision (LP) sensors, the error can reach to 2.5 meters. In this way, future work implies the study of robustness for these cases.

A possible extension of the current work might to consider the effect of obstacles, causing non-line of sight between the UAV and a (set of) reference point(s). Another interesting direction for future work might be to consider a hybrid positioning system that could merge two radio measurements together, such as RSS and TOA.

REFERENCES

[1] K. Nonami, "Prospect and recent research & development for civil use autonomous unmanned aircraft as UAV and MAV," *J. Syst. Design Dyn.*, vol. 1, no. 2, pp. 120–128, 2007.

[2] C. Yan, L. Fu, J. Zhang, and J. Wang, "A comprehensive survey on UAV communication channel modeling," *IEEE Access*, vol. 7, pp. 107769–107792, 2019.

[3] T. Tomic, K. Schmid, P. Lutz, A. Dömel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 46–56, Sep. 2012.

[4] L. López, N. Van Manen, E. Van der Zee, and S. Bos, *DroneAlert: Autonomous Drones for Emergency Response*. Cham, Switzerland: Springer, Jan. 2017, pp. 303–321.

[5] N. Macoir, J. Bauwens, B. Jooris, B. Van Herbruggen, J. Rossey, J. Hoebeke, and E. De Poorter, "UWB localization with battery-powered wireless backbone for drone-based inventory management," *Sensors*, vol. 19, no. 3, p. 467, Jan. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/3/467

[6] W. Kwon, J. H. Park, M. Lee, J. Her, S.-H. Kim, and J.-W. Seo, "Robust autonomous navigation of unmanned aerial vehicles (UAVs) for warehouses' inventory application," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 243–249, Jan. 2020.

[7] A. Mashood, A. Dirir, M. Hussein, H. Noura, and F. Awwad, "Quadrotor object tracking using real-time motion sensing," in *Proc. 5th Int. Conf. Electron. Devices, Syst. Appl. (ICEDSA)*, Dec. 2016, pp. 1–4.

[8] M. Beul, D. Droeschel, M. Nieuwenhuisen, J. Quenzel, S. Houben, and S. Behnke, "Fast autonomous flight in warehouses for inventory applications," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3121–3128, Oct. 2018.

[9] L. Campos-Macías, R. Aldana-López, R. Guardia, J. I. Parra-Vilchis, and D. Gómez-Gutiérrez, "Autonomous navigation of MAVs in unknown cluttered environments," *J. Field Robot.*, vol. 38, no. 2, pp. 307–326, May 2020, doi: 10.1002/rob.21959.

[10] E. Welburn, H. H. Khalili, A. Gupta, S. Watson, and J. Carrasco, "A navigational system for quadcopter remote inspection of offshore substations," in *Proc. 15th Int. Conf. Autonomic Auto. Syst.*, Jun. 2019, pp. 2–6.

[11] I. Kalinov, E. Safronov, R. Agishev, M. Kurenkov, and D. Tsetserukou, "High-precision UAV localization system for landing on a mobile collaborative robot based on an IR marker pattern recognition," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–6.

[12] X. Pan, C. Yan, and J. Zhang, "Joint range estimation using single carrier burst signals for networked UAVs," *IEEE Access*, vol. 9, pp. 42533–42542, 2021.

[13] G. Croon and C. De Wagter, "Challenges of autonomous flight in indoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2018, pp. 1003–1009.

[14] B. Ben-Moshe, N. Shv, J. Baadani, I. Nagar, and H. Levi, "Indoor positioning and navigation for micro UAV drones—Work in progress," in *Proc. IEEE 27th Conv. Elect. Electron. Eng. Isr.*, Nov. 2012, pp. 1–5.

[15] J. Paredes, F. Álvarez, T. Aguilera, and J. Villadangos, "3D indoor positioning of UAVs with spread spectrum ultrasound and time-of-flight cameras," *Sensors*, vol. 18, no. 2, p. 89, Dec. 2017.

[16] S. Rehman, S. Ullah, P. Chong, S. Yongchareon, and D. Komosny, "Visible light communication: A system perspective—Overview and challenges," *Sensors*, vol. 19, no. 5, p. 1153, Mar. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/5/1153

[17] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, and A. Savvaris, "LIDAR-inertial integration for UAV localization and mapping in complex environments," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2016, pp. 649–656.

[18] E. Guerra, R. Munguía, and A. Grau, "UAV visual and laser sensors fusion for detection and positioning in industrial applications," *Sensors*, vol. 18, no. 7, 2018, Art. no. 2071. [Online]. Available: https://www.mdpi.com/1424-8220/18/7/2071, doi: 10.3390/s18072071.

[19] J. Braga, H. Campos Velho, and E. Shiguemori, "Estimation of UAV position using LiDAR images for autonomous navigation over the ocean," in *Proc. 9th Int. Conf. Sensing Technol.*, Dec. 2015, pp. 811–816.

[20] M. M. U. Chowdhury, F. Erden, and I. Guvenc, "RSS-based Q-learning for indoor UAV navigation," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Nov. 2019, pp. 121–126.

[21] R. I. Marasigan, Y. D. Austria, J. B. Enriquez, L. L. Lacatan, and R. M. Dellosa, "Unmanned aerial vehicle indoor navigation using Wi-Fi trilateration," in *Proc. 11th IEEE Control Syst. Graduate Res. Colloq. (ICSGRC)*, Aug. 2020, pp. 346–351.

[22] J. Smith and J. Abel, "Closed-form least-squares source location estimation from range-difference measurements," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 12, pp. 1661–1669, Dec. 1987.

[23] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 1905–1915, Aug. 1994.

[24] K. Doğançay and A. Hashemi-Sakhtsari, "Target tracking by time difference of arrival using recursive smoothing," *Signal Process.*, vol. 85, no. 4, pp. 667–679, Apr. 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168404003093

[25] Y. Weng, W. Xiao, and L. Xie, "Total least squares method for robust source localization in sensor networks using TDOA measurements," *Int. J. Distrib. Sensor Netw.*, vol. 7, no. 1, Jan. 2011, Art. no. 172902, doi: 10.1155/2011/172902.

[26] L. Lin, H. C. So, F. K. W. Chan, Y. T. Chan, and K. C. Ho, "A new constrained weighted least squares algorithm for TDOA-based localization," *Signal Process.*, vol. 93, no. 11, pp. 2872–2878, Nov. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016516841300145X

[27] M. Khalaf-Allah, "An extended closed-form least-squares solution for three-dimensional hyperbolic geolocation," in *Proc. IEEE Symp. Ind. Electron. Appl. (ISIEA)*, Sep. 2014, pp. 7–11.

[28] M. Khalaf-Allah, "Performance comparison of closed-form least squares algorithms for hyperbolic 3-D positioning," *J. Sensor Actuator Netw.*, vol. 9, no. 1, p. 2, Dec. 2019. [Online]. Available: https://www.mdpi.com/2224-2708/9/1/2

[29] T. Li, A. Ekpenyong, and Y.-F. Huang, "Source localization and tracking using distributed asynchronous sensors," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3991–4003, Oct. 2006.

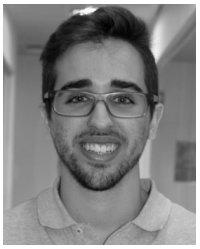
[30] M. Win, A. Conti, S. Mazuelas, Y. Shen, W. Gifford, D. Dardari, and M. Chiani, "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, May 2011.

[31] J. C. Chen, K. Yao, and R. E. Hudson, "Source localization and beamforming," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 30–39, Mar. 2002.

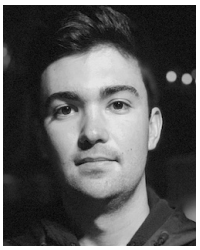
[32] K. Lui, F. Chan, and H. C. So, "Semidefinite programming approach for range-difference based source localization," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1630–1633, Apr. 2009.

[33] M. R. Gholami, S. Gezici, and E. G. Strom, "Improved position estimation using hybrid TW-TOA and TDOA in cooperative networks," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3770–3785, Jul. 2012.

[34] J. Tiemann and C. Wietfeld, "Scalable and precise multi-UAV indoor navigation using TDOA-based UWB localization," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–7.
[35] M. Khalaf-Allah, "Particle filtering for three-dimensional TDoA-based positioning using four anchor nodes," *Sensors*, vol. 20, no. 16, p. 4516, Aug. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/16/4516
[36] Y. Shen, B. Hwang, and J. P. Jeong, "Particle filtering-based indoor positioning system for beacon tag tracking," *IEEE Access*, vol. 8, pp. 226445–226460, 2020.
[37] A. M. Romero. (2014). *Ros/Concepts*. [Online]. Available: http://wiki.ros.org/ROS/Concepts
[38] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 398–409, Apr. 2015.
[39] D. Pedro, J. P. Matos-Carvalho, F. Azevedo, R. Sacoto-Martins, L. Bernardo, L. Campos, J. M. Fonseca, and A. Mora, "FFAU—Framework for fully autonomous UAVs," *Remote Sens.*, vol. 12, no. 21, p. 3533, Oct. 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/21/3533
[40] ROS. (2010). *Std_MSGS/Empty Message*. [Online]. Available: http://docs.ros.org/en/melodic/api/std_msgs/html/msg/Empty.html
[41] AnisKoubaa. (2010). *Services*. [Online]. Available: http://wiki.ros.org/Services
[42] S. Ashraf, P. Aggarwal, P. Damacharla, H. Wang, A. Y. Javaid, and V. Devabhaktuni, "A low-cost solution for unmanned aerial vehicle navigation in a global positioning system–denied environment," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 6, Jun. 2018, Art. no. 155014771878175.
[43] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," *Intell. Auton. Syst.*, vol. 302, pp. 335–348, Jan. 2016.
[44] Vooon. (2018). *Mavros*. [Online]. Available: http://wiki.ros.org/mavros
[45] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Sci. Robot.*, vol. 5, no. 40, Mar. 2020, Art. no. eaaz9712. [Online]. Available: https://robotics.sciencemag.org/content/5/40/eaaz9712
[46] D. Project. (2010). *Mavlink Developer Guide*. [Online]. Available: https://mavlink.io/en/
[47] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, 1st ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[48] J. P. Beaudeau, M. F. Bugallo, and P. M. Djuric, "RSSI-based multi-target tracking by cooperative agents using fusion of cross-target information," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5033–5044, Oct. 2015.
[49] D. Dardari, P. Closas, and P. M. Djuric, "Indoor tracking: Theory, methods, and technologies," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1263–1278, Apr. 2015.
[50] E. Masazade, R. Niu, and P. K. Varshney, "Dynamic bit allocation for object tracking in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5048–5063, Oct. 2012.
[51] G. Wang, Y. Li, and M. Jin, "On MAP-based target tracking using range-only measurements," in *Proc. 8th Int. Conf. Commun. Netw. China (CHINACOM)*, Guilin, China, Aug. 2013, pp. 1–6.
[52] J. J. Moré, "Generalizations of the trust region problem," *Optim. Meth. Soft.*, vol. 2, nos. 3–4, pp. 189–209, Feb. 1993.
[53] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1770–1778, May 2008.
[54] S. Tomic, M. Beko, and R. Dinis, "3-D target localization in wireless sensor networks using RSS and AoA measurements," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3197–3210, Apr. 2017.
[55] J. O'Kane, "Global localization using odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, Feb. 2006, pp. 37–42.
[56] R. Labbe. (2020). *Kalman and Bayesian Filters in Python*. [Online]. Available: https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python
[57] J. Nakama, R. Parada, J. P. Matos-Carvalho, F. Azevedo, D. Pedro, and L. Campos, "Autonomous environment generator for UAV-based simulation," *Appl. Sci.*, vol. 11, no. 5, p. 2185, Mar. 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/5/2185
[58] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Sep. 2004, pp. 2149–2154.
[59] J. P. Matos-Carvalho, D. Pedro, L. M. Campos, J. M. Fonseca, and A. Mora, "Terrain classification using W-K filter and 3D navigation with static collision avoidance," in *Intelligent Systems and Applications*, Y. Bi, R. Bhatia, and S. Kapoor, Eds. Cham, Switzerland: Springer, 2020, pp. 1122–1137.
[60] H. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "RViz: A toolkit for real domain data visualization," *Telecommun. Syst.*, vol. 60, pp. 1–9, Oct. 2015.
[61] S. Tomic, M. Beko, and R. Dinis, "RSS-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 2037–2050, May 2015.



J. P. MATOS-CARVALHO was born in Almada, Portugal, in 1993. He received the M.Sc. degree in electrical and computer engineer from FCT/UNL, Portugal, in 2017, finishing among the top 1% of the students graduating in that year, and the Ph.D. degree in electrical and computer engineering, in 2021. He has been working in the aerial robotics research and development field, since 2016. He is currently an Assistant Professor with the Universidade Lusófona de Humanidades e Tecnologias, Lisbon. He received the Best Paper Award "UAV downwash dynamic texture features for terrain classification on autonomous navigation" at the prestigious IEEE conference, in 2018.



RICARDO SERRAS SANTOS was born in Lisbon, Portugal, in 1999. He is currently pursuing the B.Sc. degree in informatics, networks, and telecommunications engineering with the Universidade Lusófona de Humanidades e Tecnologias. He started researching target localization and aerial robotics in November 2020.

SLAVISA TOMIC received the M.S. degree in traffic engineering from the University of Novi Sad, Serbia, in 2010, through the Postal Traffic and Telecommunications Study Program, and the Ph.D. degree in electrical and computer engineering from the University Nova of Lisbon, Portugal, in 2017. He is currently an Assistant Professor at the Universidade Lusófona de Humanidades e Tecnologias, Lisbon, Portugal. According to the methodology proposed by Stanford University, he was among the most influential researchers in the world in 2019, when he joined the list of top 1% of scientists whose work is most cited by other colleagues in the field of information and communication technologies, sub-area networks, and telecommunications. His research interests include target localization in wireless sensor networks and non-linear and convex optimization.

MARKO BEKO was born in Belgrade, Serbia, in November 1977. He received the Ph.D. degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Universidade de Lisboa, Portugal, in 2008. He received the title of Professor with Habilitation of electrical and computer engineering from the Universidade Nova de Lisboa, Lisbon, in 2018. His current research interest includes signal processing for wireless communications. He is a member of the Editorial Board of IEEE Open Journal of Vehicular Technology. He was the winner of the 2008 IBM Portugal Scientific Award. According to the methodology proposed by Stanford University, he was among the most influential researchers in the world in 2019, when he joined the list of top 1% of scientists whose work is most cited by other colleagues in the field of information and communication technologies, sub-area networks, and telecommunications. He is one of the founders of Koala Tech. He serves as an Associate Editor for the IEEE Open Journal of the Communications Society and journal on *Physical Communication* (Elsevier).