UAV Navigation in Satellite-less Environments

Ricardo Jorge Serras Santos, *MEEC*

***Abstract*—The usage of Unmanned Aerial Vehicles (UAV) has grown exponentially in recent years and are now indispensable for some applications. However, the majority of current commercial UAVs depend mainly on satellite signals to estimate their position. Despite this fact not being a problem in outdoor environments, in scenarios where these signals may be blocked, the UAV may not be able to estimate its position which can result in severe accidents. Thus, this thesis' main objective is to develop algorithms capable of navigating a UAV in this type of scenarios. The thesis proposes 3 solutions for the problem of interest. Two of them are based on statistic modelling (least squares optimization), while the remaining one is based on data driven models (neural networks). The first solution takes advantage of information extracted from received radio signals by employing existing terrestrial technology and resorts to a Generalized Trust Region Sub-Problem (GTRS) solved via the bisection procedure to estimate the UAV's position. With the estimated position it is possible to navigate the UAV by calculating the shortest distance to the destination. The second solution follows the same principal of extracting information from radio signals, although this time a Weighed Least Squares (WLS), whose solution is found in closed-form, was used to estimate the UAV's position. The third solution resorts to machine learning, more precisely, a Long-Short Term Memory (LSTM). The results obtained from the simulations performed for the three algorithms corroborate the accuracy and fast processing time when compared to other state-of-the-art methods.**

***Index Terms*—Navigation, Long Short-Term Memory (LSTM), Unmanned Aerial Vehicle (UAV), Weighted Least Squares (WLS), Generalized Trust Region Sub-Problem (GTRS)**

I. Introduction

The usage of UAVs, commonly referred to as drones, has grown in recent years in a variety of applications. Due to their mobility, miniaturization, and flexible usage, UAVs are enabling a plethora of applications in domains such as monitoring, terrain mapping [1], search and rescue [2]–[4], telecommunications [5], agriculture [6]–[10], *etc*. This type of vehicles is also expected to be essential in future smart cities. In these cities, UAVs are holding a promise of providing efficient delivery services (goods, medications, and transplantation organs) [11], dynamically deployable mobile base stations for broadband hotspot connectivity (particularly suitable for occasional events and improving network coverage) [12], [13], infrastructure inspection (buildings, bridges, tunnels, and construction sites), surveillance (traffic, crowds, border control) [14], first responder services (including earthquakes, gas leakage, and explosions) [15] and many more. Nevertheless, future prospects and ubiquity of drones in urban areas raise significant technical and social challenges in the domain of privacy, cyber security, and public safety. In this way, not only is it essential to develop a platform to integrate drones in several tasks, but it is also necessary to address their security and privacy shortcomings. Moreover, current commercially available UAVs are not fully autonomous and require human intervention to operate safely. Also, this type of vehicles depend, mainly, on satellite signals to estimate their position. In addition to these signals, more sensors can be added, like cameras or radars to add extra awareness of the surrounding environment to the UAV. Even though the usage of such signals works relatively well in outdoor environments, in scenarios where the signal may be hindered by an obstacle or completely blocked, such as inside a building for example, the UAV may not be able to estimate its position, which can lead to loss of control and consequently severe accidents. Therefore, to overcome this problem alternative methods need to be developed for these scenarios.

One possibility is the creation of a virtual sensor with an optical-flow sensor, an orientation sensor, a range sensor and a geometric camera proposed in [16]. The authors proposed the combination of the data from these sensors using a fusion algorithm. Despite being able to govern the movement of the UAV without satellite signals, the addition of the sensors to the UAV increases its mass, which can result in reduced battery life and autonomy. Other possibility is to employ Ultrasonic Positioning (ULPS) associated with a Time-Of-Flight (TOF) camera [17]. In spite of achieving relatively good results, the biggest limitation of this method is the necessity to install the ULPS on the ceiling of the room and the necessity to connect the TOF camera to a computer to be able to estimate the UAV's position. Furthermore, the UAV also requires the installation of a acoustic receiver increasing its mass. In [18], the authors proposed the usage of Received Signal Strength (RSS) acquired at the UAV in combination with a Q-learning algorithm. The limitation of this approach is that it assumes that the source of the Radio Frequency (RF) signal is stationary. In [19] the authors proposed the use of Wireless-Fidelity (Wi-Fi) signals and developed a trilateration algorithm to accomplish UAV navigation in indoor environments. In [20], an approach using Bluetooth Low Energy (BLE) beacon messages and a Particle Filter (PF) was proposed. First, a Kalman Filter (KF) was applied to preprocess collected RSS information in order to smooth the fluctuated RSS data. Afterwards, the PF is applied to approximate the unknown position of a tag and reduce the position uncertainties in a Gaussian belief space and RSS gradient model, for the motion estimation. An advantage of this approach is that it does not require the use of additional sensors on the UAV. However, the PF performance depends highly on the number of particles employed. The use of Simultaneous Localisation and Mapping (SLAM) is another possibility proposed in [21]. This method uses Red Green Blue-Depth (RGBD)-based Visual-Inertial Odometry, a semantic segmentation and 3D reconstruction in real-time to create a 3D semantic map that can be used for autonomous UAV navigation. The main drawback is, such as in some of the previous methods, the installation of additional sensors

on the UAV and the necessary computational power needed to create a 3D map in real-time leading to a reduced battery life and autonomy. More recently, artificial intelligence and machine learning algorithms have also been proposed. In [22], the authors proposed a deep neural network composed by a Convolutional Neural Network (CNN) that excerpts the local spatial features from the measurements, followed by a deep LSTM model that draws out temporal features. Even though high accuracy can be achieved by such methods, they have limited applicability in highly dynamic environments, since they require a training phase.

II. Problem Formulation

Taking into consideration a 3D wireless network with a set of N stationary reference points, whose positions are denoted by $\boldsymbol{a}_i = [a_{1,i}, a_{2,i}, a_{3,i}]^T$, $i = 1, ..., N$, and a UAV whose unknown position at any time instant, t, is denoted by $\boldsymbol{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}, x_3^{(t)}]^T$. Assuming that, at every instant t, a set of radio signals is transmitted between the UAV and a reference point, it is possible to extract distance information between them (for instance, from TOF [23], [24] or RSS [25], [26] information). Therefore, the k-th sample distance measurement between the UAV and i-th reference point at time instant t is given by

$$d_{i,k}^{(t)} = \|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\| + n_{i,k}^{(t)}, \;\; i = 1, ..., N, \;\; k = 1, ..., K, \quad (1)$$

where the symbol $\| \bullet \|$ denotes the Euclidean norm and $n_{i,k}^{(t)}$ designates the measurement noise, considered as a zero-mean normal random variable with standard deviation $\sigma_{i,k}^{(t)}$, *i.e.*, $n_{i,k}^{(t)} \sim \mathcal{N}(0, \sigma_{i,k}^{(t)})$. In order to introduce more robustness to outliers and simplify the notation, the median distance, $d_i^{(t)}$, from the total K measurement samples is calculated and used to estimate the UAV's position at time instant t.

Based on the measurements in (1), the ML estimator [27] of $\boldsymbol{x}$ at time instant t is found by solving the following optimization problem.

$$\widehat{\boldsymbol{x}}^{(t)} = \underset{\boldsymbol{x}^{(t)}}{\arg\min} \sum_{i=1}^{N} \frac{1}{2(\sigma_i^{(t)})^2} \left(d_i^{(t)} - \|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\| \right)^2. \quad (2)$$

The ML estimator is often used in practice, since it is asymptotically efficient (for large enough data records). Nevertheless, note that (2) is non-convex (due to the norm term in the measurement model) and cannot be tackled directly. Therefore, to bypass this problem, this thesis introduces three approaches to obtain (a sub-optimal) estimator of (2). One is to convert the problem into a GTRS that can be solved via the bisection procedure. The second one is to cast the ML into a WLS solved in closed form, while the third one is based on the use of deep learning tools, more specifically an LSTM. In the following three chapters, the three methods are explained in detail.

Since the main goal of this thesis is to accurately navigate the UAV and knowing its position, say via (2) one can formulate the navigation problem, in 3-D, using some intermediate points in the area of interest as

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{S}\boldsymbol{\theta}^{(t)}, \quad (3)$$

where $\boldsymbol{\theta}^{(t)} = \left[(\boldsymbol{x}^{(t)})^T, (\boldsymbol{v}^{(t)})^T \right]^T$ denotes the state of the UAV at time t, $\boldsymbol{x}^{(t)}$ represents the UAV's position at time t, $\boldsymbol{S}$ is the state transition matrix represented by

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 & 0 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

with Δ being the sampling interval between two following time steps, while

$$\boldsymbol{v}^{(t)} = \boldsymbol{\nu}^{(t)} \begin{bmatrix} \cos(\varphi^{(t)}) \sin(\varrho^{(t)}) \\ \sin(\varphi^{(t)}) \sin(\varrho^{(t)}) \\ \cos(\varrho^{(t)}) \end{bmatrix} \in \mathbb{R}^3 \quad (5)$$

denotes the velocity vector of the UAV, with $\varphi^{(t)}$ and $\varrho^{(t)}$ representing respectively, the azimuth angle and the elevation angle between the current UAV position and the desired destination. Notice that, in general, $\boldsymbol{\nu}^{(t)}$ in (5) is given by

$$\boldsymbol{\nu}^{(t)} = \boldsymbol{\omega}^{(t)} \cdot P_{vmax}\,, \quad (6)$$

where P_{vmax} is the maximum speed at which the UAV can fly. However, when the UAV is closer than τ meters to an intermediate waypoint (destination), $\boldsymbol{\nu}^{(t)}$ in (5) is obtained using

$$\boldsymbol{\nu}^{(t)} = \boldsymbol{\eta}^{(t)} \cdot \left(\frac{\lambda^{(t)}}{\tau} \right)^{\gamma}, \quad (7)$$

where $\boldsymbol{\eta}^{(t)} = \boldsymbol{x}_{\text{dest}} - \boldsymbol{x}^{(t)}$, $\lambda^{(t)} = \|\boldsymbol{\eta}^{(t)}\|$ denotes the distance between the current UAV position and the desired destination, τ is a threshold that evaluates the proximity of the UAV to the current waypoint and γ is a smoothing factor.

III. The Proposed GTRS-based Scheme for Drone Navigation

A. Estimating the UAV's position

This solution proposes to overcome the ML estimator in (2) by transforming it into a GTRS optimization problem. To do so, one starts with the introduction of weights $\boldsymbol{w}^{(t)} = [w_i^{(t)}]^T$, where

$$w_i^{(t)} = \frac{\left(d_i^{(t)}\right)^{-1}}{\sum_{i=1}^{N} \left(d_i^{(t)}\right)^{-1}} \quad (8)$$

is defined in order to assign more trust to closer links. Resorting to (1) and applying the WLS criterion, the ML can be approximated by another non-convex estimator as

$$\underset{\boldsymbol{x}^{(t)}}{\text{minimize}} \sum_{i=1}^{N} w_i \left(\left(d_i^{(t)}\right)^2 - \|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\|^2 \right)^2, \quad (9)$$

By developing the squared-norm terms in (9), it is possible to write it in vector notation as

$$\begin{aligned} &\underset{\boldsymbol{y}^{(t)}}{\text{minimize}} \;\; \|\boldsymbol{A}^{(t)}\boldsymbol{y}^{(t)} - \boldsymbol{b}^{(t)}\|^2 \\ &\text{subject to} \\ &(\boldsymbol{y}^{(t)})^T \boldsymbol{D}\boldsymbol{y}^{(t)} + 2\boldsymbol{f}^T\boldsymbol{y}^{(t)} = 0, \end{aligned} \quad (10)$$

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

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

where $\boldsymbol{I}_G$ and $\boldsymbol{0}_{p\times q}$ denote the identity matrix of size G and the matrix of all-zero entries of size $p \times q$. By doing this transformation, an optimization problem consisting of a quadratic objective function and a quadratic constraint is achieved. This class of optimization problems is known as a GTRS [28], [29] and is generally non-linear and non-convex. Despite this property, it is possible to calculate an interval where the problem is a monotonically decreasing function. Within this interval, it is possible to solve (10) *exactly* by just the bisection procedure [29]. Hence, by solving (10), one can estimate the UAV's position at any time instant by resorting only to radio measurements.

B. Enhancing Positioning Accuracy via Accumulated Knowledge

By following Bayesian philosophy and combining the knowledge accumulated during time with the observations, a marginal posterior PDF, $p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t)})$ is created, where $\boldsymbol{\theta}^{(t)}$ represents the UAV's state by its location and velocity at time instant t, *i.e.*, $\boldsymbol{\theta}^{(t)} = \left[(\boldsymbol{x}^{(t)})^T, (\boldsymbol{v}^{(t)})^T \right]^T$.

It is assumed that the target is moving according to a nearly constant velocity motion model, defined as

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{S}\,\boldsymbol{\theta}^{(t-1)} + \boldsymbol{r}^{(t)}, \tag{11}$$

where S is the transition matrix in (4) and $\boldsymbol{r}^{(t)} = [(\boldsymbol{r}_{\boldsymbol{x}}^{(\boldsymbol{t})})^T, (\boldsymbol{r}_{\boldsymbol{v}}^{(\boldsymbol{t})})^T]^T$ denotes the state process noise [30], [31], that follows a zero-mean Normal distribution with covariance matrix $\boldsymbol{Q}$, *i.e.*, $\boldsymbol{r}^{(t)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$. This matrix is described by

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

with q being the intensity of the state process noise [31], [32]. The presented marginal posterior is of particular interest since it quantifies the belief in the values of the state, $\boldsymbol{\theta}^{(t)}$, given all prior observations, $\boldsymbol{d}^{(1:t)}$, from which an estimate at any time instant can be obtained. A summarized version of the key parts of the Bayesian philosophy in [27] is given below.

- *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 and use the measurements up to $t-1$ to obtain the predictive density of the state at t as

$$\begin{aligned} &p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)}) \\ &= \textstyle\int p(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t-1)}) p(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{d}^{(1:t-1)}) d\boldsymbol{\theta}^{(t-1)}. \end{aligned} \tag{12}$$

- *Update:* Apply Bayes' rule [31], [33], to get

$$p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t)}) = \frac{p(\boldsymbol{d}^{(t)}|\boldsymbol{\theta}^{(t)}) p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)})}{p(\boldsymbol{d}^{(t)}|\boldsymbol{d}^{(1:t-1)})}, \tag{13}$$

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 only required to assure that $p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)})$ integrates to 1 [27].

According to (13) and the MAP criterion [27], an estimator of $\boldsymbol{\theta}^{(t)}$ can be obtained as

$$\begin{aligned} \hat{\boldsymbol{\theta}}^{(t)} &= \arg\max_{\boldsymbol{\theta}^{(t)}} \; p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t)}) \\ &= \arg\max_{\boldsymbol{\theta}^{(t)}} \; p(\boldsymbol{d}^{(t)}|\boldsymbol{\theta}^{(t)})\, p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t-1)}). \end{aligned} \tag{14}$$

Note that the density in (12) 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 (12) has no closed-form solution. Thus, some approximations are usually necessary to find $p(\boldsymbol{\theta}^{(t)}|\boldsymbol{d}^{(1:t)})$.

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 (12), 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, defined as

$$c = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \tag{16}$$

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

$$\hat{\boldsymbol{\theta}}^{(t|t-1)} = \boldsymbol{S}\hat{\boldsymbol{\theta}}^{(t-1|t-1)}, \tag{17a}$$

$$\hat{\boldsymbol{P}}^{(t|t-1)} = \boldsymbol{S}\hat{\boldsymbol{P}}^{(t-1|t-1)}\boldsymbol{S}^T + \boldsymbol{Q}. \tag{17b}$$

By using (10) and (15), we can then approximate (14) as

$$\begin{aligned} &\underset{\boldsymbol{z}^{(t)}}{\text{minimize}} \; \|\widetilde{\boldsymbol{A}}^{(t)}\boldsymbol{z}^{(t)} - \widetilde{\boldsymbol{b}}^{(t)}\|^2 \\ &\text{subject to} \\ &(\boldsymbol{z}^{(t)})^T \widetilde{\boldsymbol{D}} \boldsymbol{z}^{(t)} + 2\widetilde{\boldsymbol{f}}^T \boldsymbol{z}^{(t)} = 0, \end{aligned} \tag{18}$$

[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).

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

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

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

and

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

The problem in (18) is a GTRS, and similarly as (10), it can be solved *exactly* via a bisection method.

C. Enhancing Positioning Accuracy via Odometry

To further enhance the position accuracy, a position estimation is obtained by using the UAV's transform in relation to the world using information from the UAV's sensors. This can be defined as

$$\hat{\boldsymbol{x}}_{\text{Odom}}^{(t)} = h(\boldsymbol{acc}^{(\boldsymbol{t})}, \boldsymbol{gy}^{(\boldsymbol{t})}), \tag{19}$$

where $\boldsymbol{acc}^{(\boldsymbol{t})}$ is the 3-axis accelerometer, $\boldsymbol{gy}^{(\boldsymbol{t})}$ is the 3-axis gyroscope and h represents the transformation [34] in order to discover the local position of the UAV (in this thesis, it is assumed that $\hat{\boldsymbol{x}}_{\text{Odom}}^{(t)}$ is equal to 0 when the positioning algorithm is launched).

Note that relying only on odometry might not be a good idea in practice, since any external force, such as wind gusts, acting on the UAV will likely result in a erroneous position estimation. Nonetheless, this information can be exploited to complement and further enhance the accuracy of the GTRS algorithm.

To take advantage of these two pieces of information and enhance the positioning accuracy of the GTRS algorithm, a Kalman filter [35] is used to determine the final UAV position at time instant t as

$$\hat{\boldsymbol{x}}^{(t)} = KF.predict(\hat{\boldsymbol{x}}_{\text{GTRS}}^{(t)},\ \hat{\boldsymbol{x}}_{\text{Odom}}^{(t)}). \tag{20}$$

This type of filter is often used to estimate parameters that are not measured or observed with great accuracy.

D. Navigating the UAV

Having the UAV's position, $\hat{\boldsymbol{x}}^{(t)}$, and the destination point, $\boldsymbol{x}_{\text{dest}}$, which is known beforehand, to govern the movement of the UAV one starts by estimating the direction that the UAV needs to follow by using

$$\hat{\varphi}^{(t)} = \arctan\left(\frac{x_{\text{dest},2} - \hat{x}_2^{(t)}}{x_{\text{dest},1} - \hat{x}_1^{(t)}}\right) \tag{21a}$$

and

$$\hat{\varrho}^{(t)} = \arccos\left(\frac{x_{\text{dest},3} - \hat{x}_3^{(t)}}{\|\boldsymbol{x}_{\text{dest}} - \hat{\boldsymbol{x}}^{(t)}\|}\right), \tag{21b}$$

where $\hat{\varphi}^{(t)}$ and $\hat{\varrho}^{(t)}$ denote, respectively, the azimuth angle and the elevation angle from the estimated position, $\hat{\boldsymbol{x}}^{(t)}$, to a desired destination, at time instant t.

Now, the subsequent state can be obtained using

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \boldsymbol{S}\hat{\boldsymbol{\theta}}^{(t)}, \tag{22}$$

where $\hat{\boldsymbol{\theta}}^{(t)} = \left[(\hat{\boldsymbol{x}}^{(t)})^T, (\hat{\boldsymbol{v}}^{(t)})^T\right]^T$ denotes the estimated state at time t, $\boldsymbol{S}$ denotes the state transition matrix defined in (4) and

$$\hat{\boldsymbol{v}}^{(t)} = \boldsymbol{\nu}^{(\boldsymbol{t})} \begin{bmatrix} \cos(\hat{\varphi}^{(t)})\sin(\hat{\varrho}^{(t)}) \\ \sin(\hat{\varphi}^{(t)})\sin(\hat{\varrho}^{(t)}) \\ \cos(\hat{\varrho}^{(t)}) \end{bmatrix} \in \mathbb{R}^3 \tag{23}$$

is the estimated velocity vector, with $\boldsymbol{\nu}^{(\boldsymbol{t})}$ defined by (7), where an estate of $\boldsymbol{\eta}$, $\hat{\boldsymbol{\eta}}^{(\boldsymbol{t})} = \boldsymbol{x}_{\text{dest}} - \hat{\boldsymbol{x}}^{(t)}$, was used and accordingly $\hat{\lambda}^{(t)} = \|\hat{\boldsymbol{\eta}}^{(\boldsymbol{t})}\|$ is the distance between the destination and the current position estimate.

Note that, if $\hat{\lambda}^{(t)} \geq \tau$, then maximum flying velocity (7) is permitted to the UAV. Otherwise, the UAV is decelerated according to (6).

E. Performance Results

To present the numerical results, some crucial parameters need to be described. The parameter values used in the simulations can be seen in Table I, while the positions of the reference points are shown in Table II.

TABLE I
PARAMETERS USED IN THE SIMULATIONS.

Parameter	Abbreviation	Value
Number of Mission Waypoints	Mw	52
Number of Monte Carlo	Mc	100
Area border	B	200 (m)
Distance threshold	τ	4 (m)
Maximum velocity	P_{vmax}	2 (m/s)
Smoothing factor	$\boldsymbol{\gamma}$	2 according to [36]

TABLE II
TRUE LOCATIONS OF THE REFERENCE POINTS USED IN THE SIMULATION ENVIRONMENT.

i	1	2	3	4	5	6	7	8
$a_{1,i}$	0	0	$B/2$	$B/2$	0	0	$B/8$	0
$a_{2,i}$	0	$B/2$	0	$B/2$	$B/10$	$B/12$	0	$B/4$
$a_{3,i}$	$B/4$	$B/4$	$B/4$	$B/4$	$B/4$	$B/4$	$B/4$	$B/4$

To provide a better comprehension of the performance of the algorithm, the main metric used is the Root Mean Square Error (RMSE) [26], and 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. Furthermore, 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. Finally, 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.

The influence of N is represented in Figure 1, where, it is possible to conclude that as N increases, the estimation accuracy betters in general. This is due to the fact that with more reference points, a higher quantity of radio measurements are obtained and used by the algorithm. Also, the symmetry of the network seems to influence the results of the algorithm. Note that, for instance, when N = 4, the algorithm reveals a superior performance when compared to N = 7. This observation shows that adding more anchors does not always results in a superior performance.

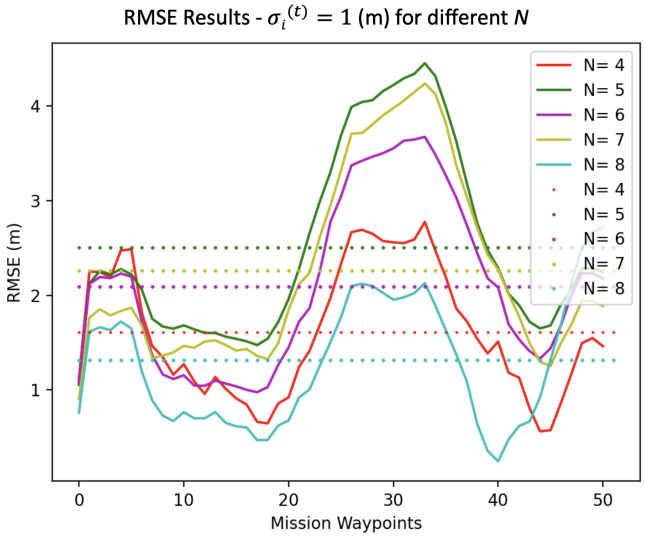


Fig. 1. Simulation results for the GTRS based algorithm with $\sigma_i^{(t)}$ = 1 (m) $\forall i, \forall t$ and N values between 4 and 8, with increments of 1.

In Figure 2 is shown the effect of the variation of $\sigma_i^{(t)}$. The figure confirms that when the noise power is higher, the accuracy deteriorates, as expected. The impact of different K values was the next parameter that was tested. In all the simulations employed when testing this parameter, a value of $\sigma_i^{(t)}$ = 2 (m) was used, as can be seen in Figure 3. From the resulting figure, it is noticeable that when the value of K increases the positioning error lowers, as expected, since more measurements are being taken.

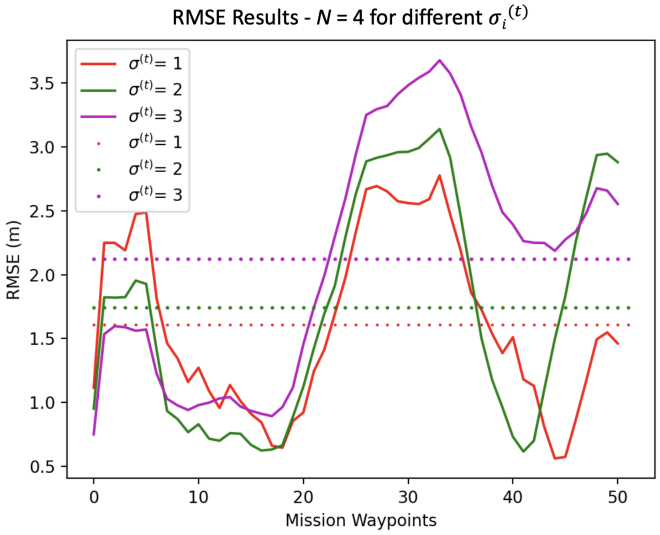


Fig. 2. Simulation results for the GTRS based algorithm with N = 4 and $\sigma_i^{(t)}$ = 1, 2 and 3 (m).

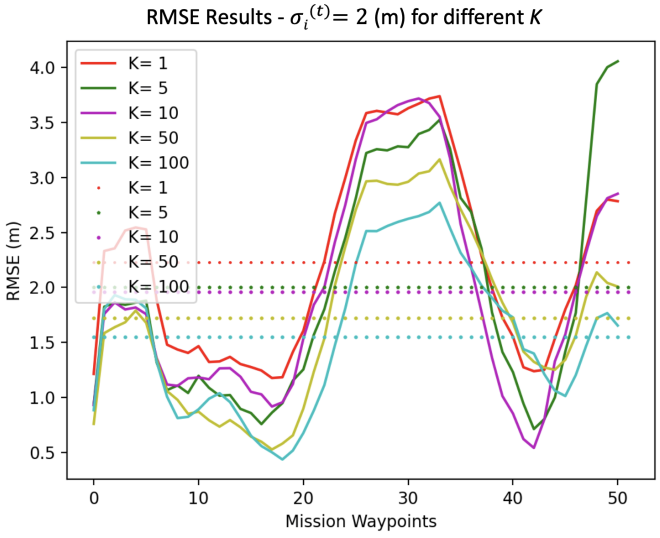


Fig. 3. Simulation results for the GTRS based algorithm with $\sigma_i^{(t)}$ = 2 (m) $\forall i, \forall t$ and K values equal to 1, 5, 10, 50 and 100.

IV. The Proposed WLS-based Scheme for Drone Navigation

A. Estimating the UAV's position

Despite the good results achieved by the GTRS algorithm, its complexity was still not the best achievable. Therefore, a second approach was implemented by casting the ML problem into a WLS.

Starting by writing the equation of a sphere centered at the i-th reference point with radius equal to its respective distance measurement to the UAV at time t as

$$\|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\|^2 = (d_i^{(t)})^2, \;\; i = 1, ..., N, \qquad (24)$$

where by developing the squared-norm term and subtracting a pair of sphere equations one from another, one gets

$$2(\boldsymbol{a}_i - \boldsymbol{a}_j)^T \boldsymbol{x}^{(t)} = (d_j^{(t)})^2 - (d_i^{(t)})^2 + \|\boldsymbol{a}_i\|^2 - \|\boldsymbol{a}_j\|^2,$$

which can be written in vector form as

$$\boldsymbol{A}\boldsymbol{x}^{(t)} = \boldsymbol{b}^{(t)}, \tag{25}$$

with

$$\boldsymbol{A} = \begin{bmatrix} \vdots \\ 2(\boldsymbol{a}_i - \boldsymbol{a}_j)^T \\ \vdots \end{bmatrix} \in \mathbb{R}^{\binom{|\mathcal{A}_i|}{2}\times 3}$$

and

$$\boldsymbol{b}^{(t)} = \begin{bmatrix} \vdots \\ (d_j^{(t)})^2 - (d_i^{(t)})^2 + \|\boldsymbol{a}_i\|^2 - \|\boldsymbol{a}_j\|^2 \\ \vdots \end{bmatrix} \in \mathbb{R}^{\binom{|\mathcal{A}_i|}{2}\times 1},$$

where the notation $\binom{n}{k}$ represents the combination (n choose k), $j \in \mathcal{A}_i$, being $\mathcal{A}_i$ the set of convenient (the meaning of *convenient* will be explained in the following text) reference points for a given reference point i and $|\bullet|$ the cardinality (the number of elements) of a set.

Resorting to (25) and the least squares criterion, each reference point i can calculate a coarse estimate of the UAV's position at time t as

$$\hat{\boldsymbol{x}}_i^{(t)} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\left(\boldsymbol{A}^T\boldsymbol{b}^{(t)}\right). \tag{26}$$

Next, the azimuth angle and the elevation angle from each of the reference points can be found by exploiting the estimates obtained above as

$$\hat{\phi}_i^{(t)} = \arctan\left(\frac{\hat{x}_{i,2}^{(t)} - a_{i,2}}{\hat{x}_{i,1}^{(t)} - a_{i,1}}\right) \tag{27}$$

and

$$\hat{\alpha}_i^{(t)} = \arccos\left(\frac{\hat{x}_{i,3}^{(t)} - a_{i,3}}{\|\hat{\boldsymbol{x}}_i^{(t)} - \boldsymbol{a}_i\|}\right) \tag{28}$$

respectively.

A unit direction vector is now formed as $\boldsymbol{u}_i^{(t)} = [\cos(\hat{\phi}_i{}^{(t)})\sin(\hat{\alpha}_i{}^{(t)}), \sin(\hat{\phi}_i{}^{(t)})\sin(\hat{\alpha}_i{}^{(t)}), \cos(\hat{\alpha}_i{}^{(t)})]^T$ to bypass the problematic norm term in (2). Resorting to the switching to spherical coordinates, $\boldsymbol{x}^{(t)} - \boldsymbol{a}_i = r_i^{(t)}\boldsymbol{u}_i^{(t)}$ leading to $\|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\| = r_i^{(t)}$ and multiplying the right-hand side of the last expression by $1 = (\boldsymbol{u}_i^{(t)})^T\boldsymbol{u}_i^{(t)}$ allows the linearization, for sufficiently low noise, of the measurement model in (1) as

$$d_i^{(t)} \approx (\boldsymbol{u}_i^{(t)})^T\left(\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\right).$$

Similar to the GTRS, by introducing weights $\boldsymbol{w}^{(t)} = [w_i^{(t)}]^T$, in a way that $w_i^{(t)} = \frac{(d_i^{(t)})^{-1}}{\sum_{i=1}^{N}(d_i^{(t)})^{-1}}$ gives more belief to *shorter* links, it is possible to cast the localization problem as a WLS problem

$$\underset{\boldsymbol{x}^{(t)}}{\text{minimize}} \sum_{i=1}^{N} w_i^{(t)}\left(d_i^{(t)} - (\boldsymbol{u}_i^{(t)})^T\left(\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\right)\right)^2.$$

Writing the above expression in a vector form, one gets

$$\underset{\boldsymbol{x}^{(t)}}{\text{minimize}} \|\boldsymbol{W}^{(t)}\left(\boldsymbol{D}^{(t)}\boldsymbol{x}^{(t)} - \boldsymbol{h}^{(t)}\right)\|^2, \tag{29}$$

with $\boldsymbol{W}^{(t)} = \text{diag}(\boldsymbol{w}^{(t)})$,

$$\boldsymbol{D}^{(t)} = \left[(\boldsymbol{u}_i^{(t)})^T\right] \in \mathbb{R}^{N\times 3}$$

and

$$\boldsymbol{h}^{(t)} = \left[d_i^{(t)} + (\boldsymbol{u}_i^{(t)})^T\boldsymbol{a}_i\right] \in \mathbb{R}^{N\times 1}.$$

The solution to the WLS problem can now be found in a closed-form as

$$\begin{aligned}\hat{\boldsymbol{x}}^{(t)} = &\left((\boldsymbol{D}^{(t)})^T(\boldsymbol{W}^{(t)})^T\boldsymbol{W}^{(t)}\boldsymbol{D}^{(t)}\right)^{-1} \\ &\times\left((\boldsymbol{D}^{(t)})^T(\boldsymbol{W}^{(t)})^T\boldsymbol{W}^{(t)}\boldsymbol{h}^{(t)}\right).\end{aligned} \tag{30}$$

Similarly as in section III-D, the navigation of the drone is achieved by computing the velocity vector which is then used to govern the movement of the UAV accordingly.

B. Performance Results

Similar to the GTRS algorithm in the previous section, some parameters need to be set for the simulations. In Table III it is visible the values that were taken into account in the succeeding figures, while on Table IV the positions of the reference points are shown.

TABLE III
PARAMETERS USED IN THE SIMULATIONS.

Parameter	Abbreviation	Value
Number of Mission Waypoints	Mw	36
Number of Monte Carlo	Mc	100
Area border	B	200 (m)
Distance threshold	τ	4 (m)
Maximum velocity	P_{vmax}	2 (m/s)
Smoothing factor	$\boldsymbol{\gamma}$	2 according to [36]

TABLE IV
TRUE POSITIONS OF THE REFERENCE POINTS USED IN THE SIMULATION ENVIRONMENT.

i	1	2	3	4	5	6	7	8
$a_{1,i}$	0	0	$B/2$	$B/2$	0	0	$B/2$	$B/2$
$a_{2,i}$	0	B	0	B	0	B	0	B
$a_{3,i}$	0	0	0	0	$B/8$	$B/8$	$B/8$	$B/8$

In Figures 4 and 5 the ideal path, represented by a red solid line, the average true trajectory represented by a yellow solid line and the standard deviation from the ideal path of the UAV in 2-D represented by the blue areas, are shown for: $\sigma_i^{(t)}$ = 1 (m) and $\sigma_i^{(t)}$ = 2 (m), respectively, when K = 50 and N = 8. From these figures it is visible that the WLS algorithm achieves a close-to-ideal mean trajectory even in noisy environments. Note that, the only noticeable deviations from the ideal trajectory are observed when the UAV needs to change its direction.

V. The Proposed UAV Navigation via Neural Networks

A. Network Architecture

The idea behind this algorithm is to employ the usage of deep-neural networks to improve the GTRS algorithm already

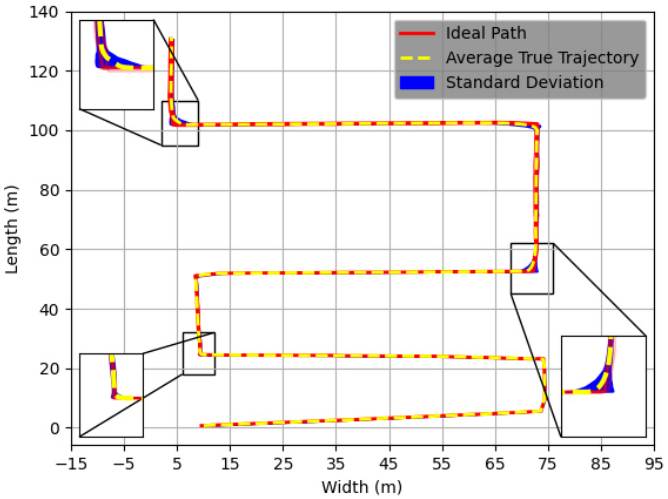


Fig. 4. Ideal trajectory vs. ground truth average estimated trajectory and standard deviation in 2-D for the WLS algorithm with $\sigma_i^{(t)}$ = 1 (m).

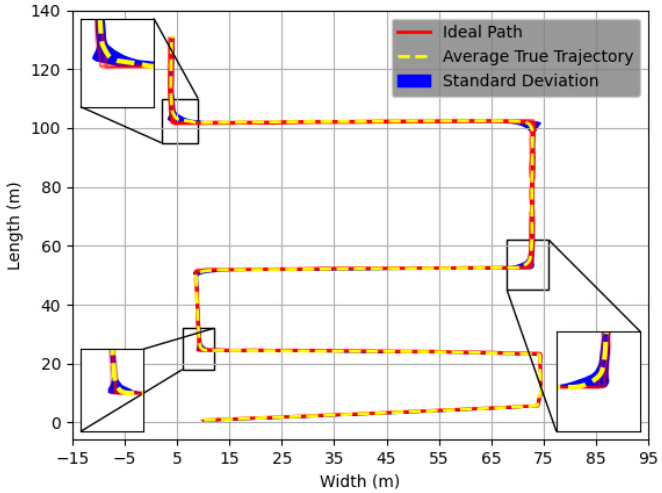


Fig. 5. Ideal trajectory vs. ground truth average estimated trajectory and standard deviation in 2-D for the WLS algorithm with $\sigma_i^{(t)}$ = 2 (m).

explained in Section III. To do so, a simple Long Short-Term Memory is trained by a dataset of GTRS estimations and then used to make its own prediction in 2-D.

In this case, the custom recurrent neural network is composed of a hidden layer (LSTM) with 4 dimensions with one Feed-forward Neural Network (FNN) visible layer (output) with one neuron, as illustrated in Figure 6. These parameters were chosen since the LSTM needs to be able to quickly process the waypoint predictions in order to be suitable for real-time scenarios.

The output of each hidden layer is activated by the sigmoid function defined as

$$\sigma(\lambda) = \frac{1}{1 - \exp\left[-\lambda\right]}, \tag{31}$$

while the last layer (the output layer) is activated by a Rectified Linear Unit (ReLU) function, defined as

$$\boldsymbol{r}(x) = max(0, x). \tag{32}$$

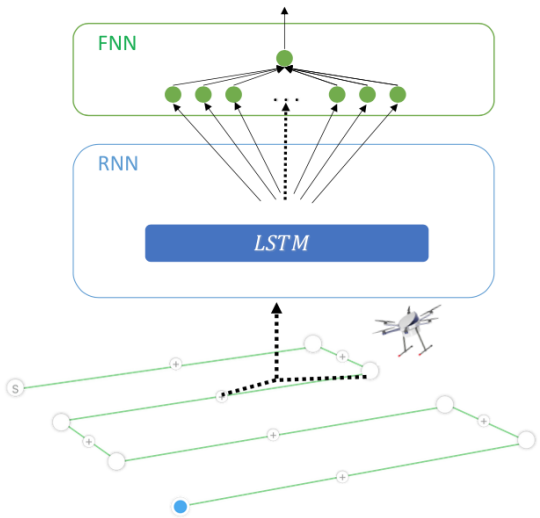


Fig. 6. The proposed neural network pipeline for the waypoints prediction.

Table V summarizes all layers of the network and its hyperparameters, where N_n denotes the amount of neurons, $optm$, represents the optimizer used to compile the model and $loss$ denotes the loss function used. The values presented in the table were obtained, by using the summary function available from Keras and Tensorflow libraries for the presented network.

TABLE V
THE PROPOSED CUSTOM RNN ARCHITECTURE HYPERPARAMETERS.

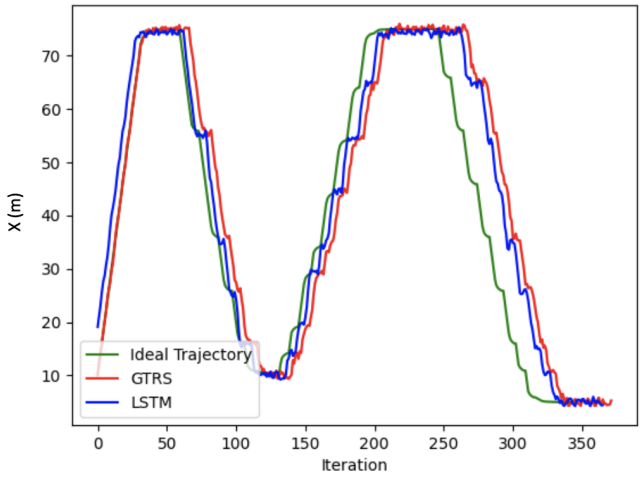
Layer	Output	Hyperparameters
INPUT	4	-
LSTM	4	N_n = 320;
FC OUTPUT	1	N_n = 5; $optm$ = ADAM $loss$ = mean_squared_error

B. Performance Results

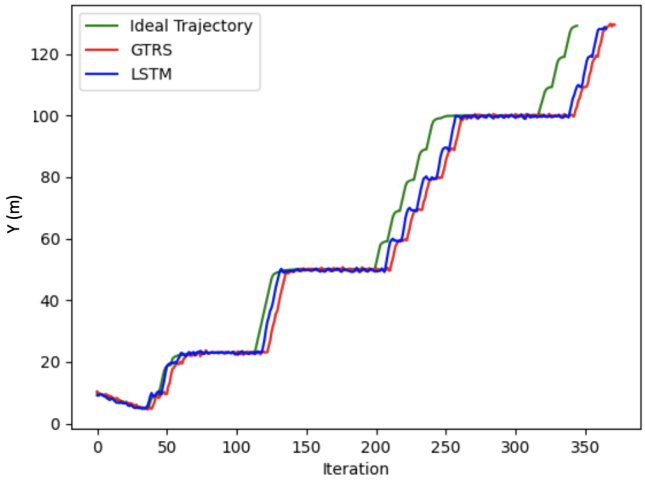
To study how the LSTM would impact the already existing GTRS algorithm in [37], it is important to define some new parameters that were added during this process. Hence, starting with the $LookBack$ (LB) that defines how many previous time instants are used to make a prediction for the subsequent time interval. Next, is the number of $Epochs$ (Eps) which defines the number of times that the learning algorithm will work through the entire training dataset. Lastly, the $TrainingPercentage$ (TP) that defines how many observations of the dataset are taken into consideration to train the model. In all simulations, the training of the LSTM uses a dataset of GTRS estimations where $\sigma_i^{(t)}$ is set to 1 (m) and Eps = 25. This dataset is used to create a model that is then used to calculate a trajectory prediction for $\sigma_i^{(t)}$ = 5 (m).

Figures 7 and 8 show the results for a TP of 100% and LB of 5 and 15, respectively, for the X- and Y- axes of a Cartesian coordinate system. On the one hand, one can see from 7 and 8 that when LB = 5 the difference between the GTRS and LSTM predictions is almost nonexistent. This is

explained by the fact that the number of previous time steps being used is not sufficiently large to make an impact on the predictions, as anticipated. On the other hand, when LB = 15, the predictions made by the LSTM are more accurate than the GTRS method. This result indicates that the LSTM is less susceptible to noise than the GTRS. Note, however, that the initial estimation of the LSTM is in some cases worse than the GTRS one. This is due to the value that is attributed to the LB parameter, *i.e.*, the LSTM only makes a prediction after LB iterations.



(a) Mission in X-axis

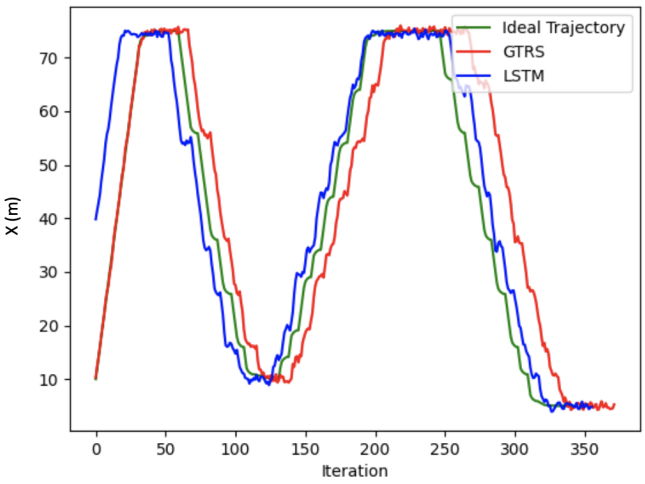


(b) Mission in Y-axis

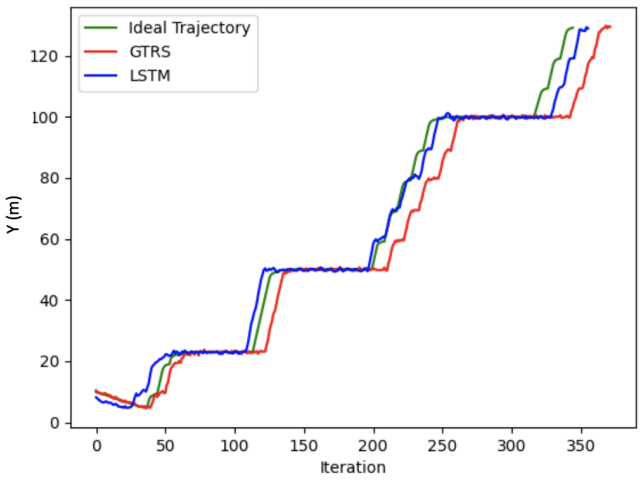
Fig. 7. Comparison between the GTRS, the LSTM and the desired trajectory of the UAV with LB = 5 and a TP of 100%.

The impact of TP on the LSTM's performance was also studied. Figures 9 and 10 show the obtained results. From them, it is possible to conclude that with more observations taken into consideration in each dataset, the precision of the LSTM trajectory enhances, as expected. Moreover, the results indicate that the LSTM does not need to use the full dataset (the whole trajectory) in order to outperform the GTRS algorithm.

Since the proposed algorithm is expected to work in real-time scenarios, the time of execution for both the GTRS and LSTM algorithms was also subject of analysis. The results obtained are shown in Table VI. This table presents the average time needed to execute one Monte Carlo run (one whole trajectory) for each method. The table shows that, in this case, the inclusion of the LSTM only added 0.01 (s) to the execution time while achieving better accuracy. The table also reveals that both algorithms are well-suited for real-time implementation.



(a) Mission in X-axis



(b) Mission in Y-axis

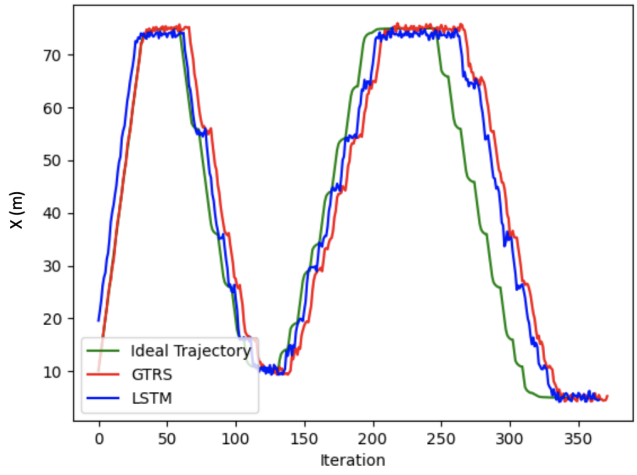
Fig. 8. Comparison between the GTRS, the LSTM and the desired trajectory of the UAV with LB = 15 and a TP of 100%.

TABLE VI
AVERAGE EXECUTION TIME FOR THE GTRS AND LSTM.

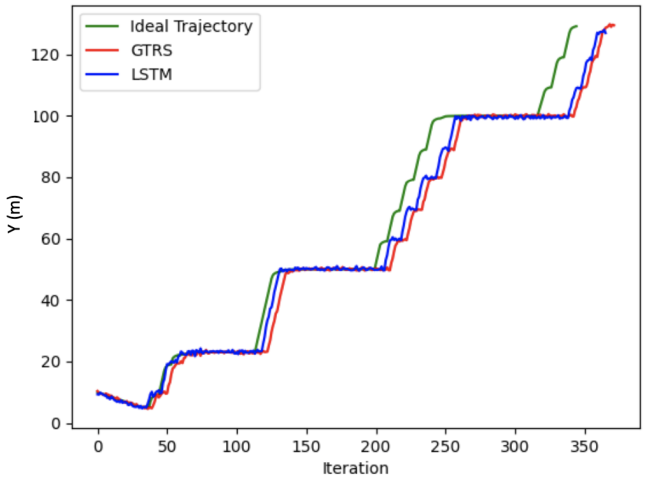
	Execution time
GTRS	0.90 (s)
LSTM	0.91 (s)

VI. CONCLUSION AND FUTURE WORK

This work studied the problem of UAV navigation in satellite-less environments. Since in such environments the

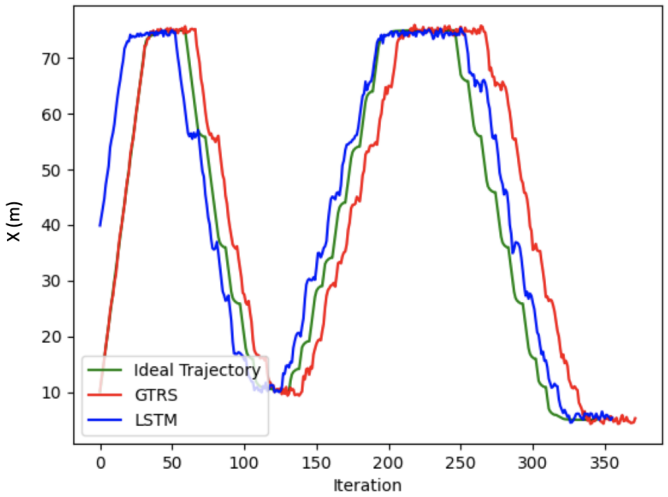


(a) Mission in X-axis

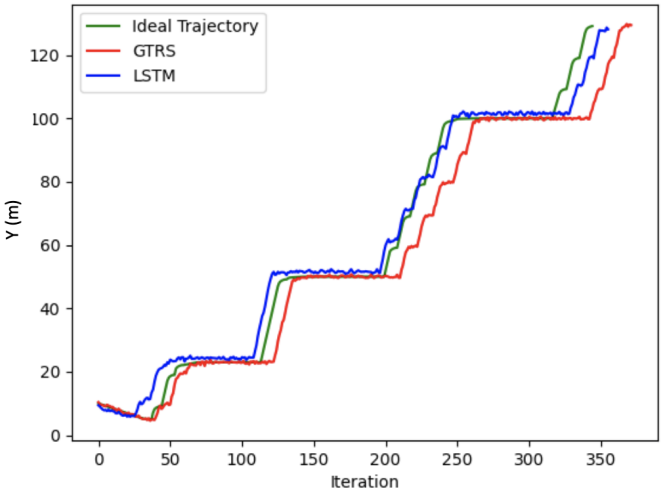


(b) Mission in Y-axis

Fig. 9. Comparison between the GTRS prediction, the LSTM prediction and the ground truth position of the UAV with LB = 5 and a TP of 70%.



(a) Mission in X-axis



(b) Mission in Y-axis

Fig. 10. Comparison between the GTRS prediction, the LSTM prediction and the ground truth position of the UAV with LB = 15 and a TP of 70%.

UAV is not able to receive GPS signals, its ability to navigate autonomously is compromised. Thus, alternative (terrestrial) solutions are required to govern the movement of the UAV. The research started by the problem formulation relying on an ML estimator. However, this type of estimator is non-convex and cannot be tackled directly. Therefore, three novel algorithms based on GTRS, WLS and LSTM frameworks to circumvent the difficulties of the considered problem were presented. These solutions were published in 2 peer-reviewed international journals and 1 international conference [37]–[39].

Regarding the GTRS-based solution, it overcame the ML problem by transforming it into a GTRS that can be solved via the bisection procedure. In this way, the UAV's position can be estimated and its possible to navigate the UAV by calculating the shortest distance to the destination. The results presented reveal the effectiveness of the proposed algorithm even in noisy environments.

Regarding the WLS, a new solution for the ML problem was found. This time it was accomplished by casting it to a WLS whose solution is found in closed-form. By extracting distance information from radio measurements, calculating the intersection points between appropriate subsets of spheres, computing both the azimuth and elevation angles and solving the WLS problem, the position of the UAV can be estimated. From the results shown it is possible to conclude that the WLS algorithm is robust against noise and achieves good accuracy when compared to the GTRS and other state-of-the-art methods.

Regarding the introduction of machine learning to improve the GTRS algorithm, a custom deep-neural-network composed of an LSTM whose training is done via a dataset of GTRS predictions was created. From the results obtained, it is possible to conclude that not only did the accuracy of the navigation increase, but the new method showed greater robustness to noise, while matching the execution time of the GTRS algorithm. This improvement can be the difference between the UAV hitting an obstacle or not.

In general, one can say that the main objectives of the thesis were achieved with success.

Future work includes the study of other methods of machine learning that can possibly be used not only to even further enhance the navigation algorithms, but could also include the integration of object avoiding systems using radio measurements to complement the proposed algorithms. The research will start by considering fixed obstacle setting (such as the ones found in warehouses) and slowly advance to more challenging dynamic environments. In parallel, another approach based on Convolutional Neural Networks is also being studied. Finally, the inclusion of secure (trustworthy) localization by detecting possibly malicious attackers or faulty sensors that may occur in one or more reference points is also being studied.

REFERENCES

[1] J. Matos-Carvalho, F. Moutinho, A. B. Salvado, T. Carrasqueira, R. Campos-Rebelo, D. Pedro, L. M. Campos, J. M. Fonseca, and A. Mora, "Static and dynamic algorithms for terrain classification in uav aerial imagery," *Remote Sensing*, vol. 11, no. 21, 2019.

[2] T. Tomić, 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," *Robotics Automation Magazine, IEEE*, vol. 19, pp. 46 –56, 09 2012.

[3] L. López, N. Van Manen, E. Van der Zee, and S. Bos, *DroneAlert: Autonomous drones for emergency response*. Springer International Publishing Switzerland, Jan. 2017, pp. 303–321.

[4] M. Pino, J. P. Matos-Carvalho, D. Pedro, L. M. Campos, and J. Costa Seco, "Uav cloud platform for precision farming," in *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2020, pp. 1–6.

[5] B. Lima, N. Fachada, R. Dinis, D. Costa, and M. Beko, "Uavnoma: A uav-noma network model under non-ideal conditions," *Journal of Open Research Software*, vol. 10, p. 9, 10 2022.

[6] F. Mohammed, A. Idries, N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "Uavs for smart cities: Opportunities and challenges," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 267–273.

[7] Y. Jin, Z. Qian, and W. Yang, "Uav cluster-based video surveillance system optimization in heterogeneous communication of smart cities," *IEEE Access*, vol. PP, pp. 1–1, 03 2020.

[8] F. Qi, X. Zhu, G. Mang, M. Kadoch, and W. Li, "Uav network and iot in the sky for future smart cities," *IEEE Network*, vol. 33, no. 2, pp. 96–101, 2019.

[9] P. Singh and A. Sharma, "An intelligent wsn-uav-based iot framework for precision agriculture application," *Computers and Electrical Engineering*, vol. 100, p. 107912, 05 2022.

[10] S. Correia, J. Fé, S. Tomic, and M. Beko, "Drones as sound sensors for energy-based acoustic tracking on wildfire environments," in *Internet of Things. Technology and Applications*, L. M. Camarinha-Matos, G. Heijenk, S. Katkoori, and L. Strous, Eds. Cham: Springer International Publishing, 2022, pp. 109–125.

[11] V. Gatteschi, F. Lamberti, G. Paravati, A. Sanna, C. Demartini, A. Lisanti, and G. Venezia, "New frontiers of delivery services using drones: A prototype system exploiting a quadcopter for autonomous drug shipments," in *2015 IEEE 39th Annual Computer Software and Applications Conference*, vol. 2, 2015, pp. 920–927.

[12] S. Sobot, M. Lukic, D. Bortnik, V. Nikic, B. Lima, M. Beko, and D. Vukobratovic, "Two-tier uav-based low power wide area networks: A testbed and experimentation study," in *2023 6th Conference on Cloud and Internet of Things (CIoT)*, 2023, pp. 85–90.

[13] B. Lima, R. Dinis, D. B. da Costa, R. Oliveira, and M. Beko, "User pairing and power allocation for uav-noma systems based on multi-armed bandit framework," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 12, pp. 13 017–13 029, 2022.

[14] Z. Zaheer, A. Usmani, E. Khan, and M. A. Qadeer, "Aerial surveillance system using uav," in *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, 2016, pp. 1–7.

[15] Z. Cook and other, "Unmanned aerial system for first responders," in *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2015, pp. 306–310.

[16] B. Ben-Moshe, N. Shv, J. Baadani, I. Nagar, and H. Levi, "Indoor positioning and navigation for micro uav drones — work in progress," 11 2012, pp. 1–5.

[17] J. Paredes, F. Álvarez, T. Aguilera, and J. M. Villadangos, "3d indoor positioning of uavs with spread spectrum ultrasound and time-of-flight cameras," *Sensors*, vol. 18, p. 89, 12 2017.

[18] M. M. U. Chowdhury, F. Erden, and I. Guvenc, "Rss-based q-learning for indoor uav navigation," 11 2019, pp. 121–126.

[19] R. Marasigan, Y. D. Austria, J. B. Enriquez, L. Lolong Lacatan, and R. M. Dellosa, "Unmanned aerial vehicle indoor navigation using wi-fi trilateration," in *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)*, 2020, pp. 346–351.

[20] Y. Shen, B. Hwang, and J. P. Jeong, "Particle filtering-based indoor positioning system for beacon tag tracking," *IEEE Access*, vol. 8, pp. 226 445–226 460, 2020.

[21] S. Wang, G. Gou, H. Sui, Y. Zhou, H. Zhang, and J. Li, "Cdsfusion: Dense semantic slam for indoor environment using cpu computing," *Remote Sensing*, vol. 14, no. 4, 2022.

[22] B. Yang, J. Li, Z. Shao, and H. Zhang, "Robust uwb indoor localization for nlos scenes via learning spatial-temporal features," *IEEE Sensors Journal*, vol. 22, no. 8, pp. 7990–8000, 2022.

[23] A. Comuniello, A. Angelis, A. Moschitta, and P. Carbone, "Using bluetooth low energy technology to perform tof-based positioning," *Electronics*, vol. 11, p. 111, 12 2021.

[24] M. Y. Omar Hashem, Khaled A. Harras, "Accurate indoor positioning using ieee 802.11mc round trip time," *Pervasive and Mobile Computing*, vol. 75, p. 101416, 05 2021.

[25] T. Rappaport, *Wireless communications - principles and practice.* Prentice Hall, 1996.

[26] S. Tomic, M. Beko, and R. Dinis, "Rss-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes," *Vehicular Technology, IEEE Transactions on*, vol. 64, pp. 2037–2050, 05 2015.

[27] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, 1st ed. Prentice Hall Upper Saddle River, NJ, USA, 1993.

[28] J. Moré, "Generalization of the trust region problem," *Optim. Meth. and Soft.*, vol. 2, no. 3-4, pp. 189–209, 02 1993.

[29] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1770–1778, 2008.

[30] J. Beaudeau, M. Bugallo, and P. Djuric, "RSSI-based multi-target tracking by cooperative agents using fusion of cross-target information," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5033–5044, 10 2015.

[31] D. Dardari, P. Closas, and P. Djuric, "Indoor tracking: Theory, methods, and technologies," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015.

[32] E. Masazade, R. Niu, and P. Varshney, "Dynamic bit allocation for object tracking in wireless sensor networks," *IEEE Trans. Sign. Process.*, vol. 60, no. 10, pp. 5048–5063, 10 2012.

[33] G. Wang, Y. Li, and M. Jin, "On map-based target tracking using range-only measurements," in *2013 8th International Conference on Communications and Networking in China (CHINACOM)*, 2013, pp. 718–723.

[34] J. O'Kane, "Global localization using odometry," vol. 2006, 02 2006, pp. 37 – 42.

[35] R. Labbe, "Kalman and bayesian filters in python," https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python, 2020.

[36] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Science Robotics*, vol. 5, no. 40, 2020.

[37] J. Matos-Carvalho, R. Santos, S. Tomic, and M. Beko, "GTRS-based algorithm for UAV navigation in indoor environments employing range measurements and odometry," *IEEE Access*, vol. 9, pp. 89 120–89 132, 2021.

[38] R. Santos, J. P. Matos-Carvalho, S. Tomic, and M. Beko, "Wls algorithm for uav navigation in satellite-less environments," *IET Wireless Sensor Systems*, vol. 12, no. 3-4, pp. 93–102, 2022.

[39] R. Santos, J. P. Matos-Carvalho, S. Tomic, M. Beko, and S. D. Correia, "Applying deep neural networks to improve uav navigation in satellite-less environments," in *2022 International Young Engineers Forum (YEF-ECE)*, 2022, pp. 63–68.