A Hybrid LSTM-based Neural Network for Satellite-less UAV Navigation

Ricardo Santos*‡, João P. Matos-Carvalho*, Slavisa Tomic*, Marko Beko*† and Sérgio D. Correia*§
*COPELABS, Universidade Lusófona, Campo Grande 376, 1749-024 Lisbon, Portugal
†Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal
‡Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal
§VALORIZA, Instituto Politécnico de Portalegre, Campus Politécnico n.10, 7300–555 Portalegre, Portugal
Correspondence: f6221@ulusofona.pt

***Abstract*—This work proposes a new algorithm to address the problem of unmanned aerial vehicle (UAV) navigation in satellite-less environments by combining machine learning with existent model-based methods. The proposed network model is trained by using the predictions of two estimators, one based on a Generalized trust region sub-problem (GTRS) framework and the other one founded on a Weighted Least Squares (WLS) principle. The solutions of these two estimators are then fed to two Long Short-Term Memories (LSTMs) to create models whose predictions are averaged to achieve the final prediction output. Our numerical results show favorable performance of the new network, obtaining improved accuracy and higher robustness to noise when compared with the individual counterparts of the network used in the training phase. Consequently, the proposed method offers safer an more reliable navigation of the UAV in satellite-less environments.**

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

I. Introduction

Unmanned aerial vehicles (UAVs), commonly referred to as drones are rapidly being adopted for a variety of application domains, especially the ones related with Smart Cities. These type of vehicles can be useful in applications ranging from wild-fire detection [1], [2], through search and rescue [3], [4], telecommunications [5] and agriculture [6]–[8]. Moreover, in future Smart Cities these vehicles can provide efficient delivery services (goods, medications, and transplantation organs) [9], deployable mobile base stations allowing broadband hotspot connectivity (particularly suitable for occasional events and improving network coverage) [10], infrastructure inspection (buildings, bridges, tunnels, and construction sites) [11], surveillance (traffic, crowds) [12], first responder services (including earthquakes, gas leakage, and explosions) [13]–[17].

Despite the wide application scope, this type of vehicles face certain technical challenges such as privacy, cyber-security and public safety issues [18]. Adding to this, current commercially available UAVs do not possess sufficient autonomy to perform operations without human supervision. Furthermore, UAVs rely heavily on the use of Global Positioning System (GPS) in order to navigate themselves. Even though GPS-driven UAVs can achieve relatively good accuracy in outdoor environments, indoor and dense urban environments where GPS signal is blocked by obstacles make this system practically useless. Therefore, in order to be able to control the movement of UAVs in such environments, alternative means need to be used.

A possible alternative to satellite signals might be to use terrestrial signals from already deployed technology. For instance, the work in [19] employed Bluetooth Low Energy (BLE) beacons to transmit signals between a moving target and a set of receivers. By acquiring Received Signal Strength (RSS) observations and levelling them using a Kalman Filter (KF), the authors in [19] were able to utilise a Particle Filter (PF) to estimate the UAV position. The authors in [20] proposed another navigation scheme by applying non-linear optimization and Bayesian methodology to transform the navigation problem into a Generalized Trust Region Sub-problem (GTRS) that can be solved via bisection procedure. To further enhance the localization accuracy, the work combined the GTRS estimation with odometry measurements by utilizing a Kalman filtering. In [21], the authors proposed a navigation algorithm that exploits radio signals and by using a Weighted Least Squares (WLS) criterion derived a solution to the positioning (navigation) problem in closed-form. Most recently, artificial intelligence and machine learning algorithms have also been used for the positioning problem [22], [23]. On the one hand, the authors in [22] proposed a deep neural network composed by a Convolutional Neural Network (CNN) use goal is to extract local spatial features from the measurements, followed by a deep Long Short-Term Memory (LSTM) model that extracts out temporal features. On the other hand, in [23], the authors proposed a deep neural network composed of an LSTM that learns from the estimates obtained by an other (model-based) solution (e.g., GTRS in [20]) to autonomously reach its own solution afterwards.

In this work, a custom deep-neural network is defined and applied to combine the GTRS [20] and the WLS [21] algorithms to train the network and achieve enhanced positioning accuracy. Resorting to individual LSTMs, one for GTRS and other for WLS, we are able to train the network for a fixed scenario with the goal of acquiring an sufficient amount of information for the network to be able to acquire its own position predictions. This is achieved by averaging the two individual LSTM predictions. In order to validate the proposed approach several tests were performed by varying different

parameters, such as, the noise power values, the percentage of trajectory used in the training phase, among others.

The main contributions are summarized as follows.

- Proposes a novel deep-neural network composed of two LSTMs whose training is done through two model-based estimators. The outputs of the two individual LSTMs are then averaged to get the final prediction on the UAV's position.
- The new method lowers the estimation error, which is accomplished by feeding the network with a higher quantity of the information that naturally leads to a better overall performance in comparison with its individual counterparts.
- The proposed network is adaptable to any existing method, allowing it to be fed by even a higher number of model-based (or other) solution, which should lead to further performance improvement.

II. Problem Formulation

Consider a q-dimensional network ($q = 2$ or 3) composed of N stationary reference (anchor) nodes and an UAV, where the true positions of all devices are represented by $\boldsymbol{a}_i$, $\boldsymbol{x}^{(t)} \in \mathbb{R}^q$, with t denoting a certain instant in time. The main goal of this work is to accomplish accurate navigation of the UAV, by using some intermediate points in the area of interest. This problem can be formally represented as

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

where $\boldsymbol{x}^{(t)}$ represents the UAV's position at time t, $\boldsymbol{S}$ is the state transition matrix defined as

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

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

$$\boldsymbol{v}^{(t)} = \boldsymbol{\nu}^{(t)} \begin{bmatrix} \cos(\varphi^{(t)})\sin(\varrho^{(t)}) \\ \sin(\varphi^{(t)})\sin(\varrho^{(t)}) \\ \cos(\varrho^{(t)}) \end{bmatrix} \in \mathbb{R}^3 \tag{2}$$

denotes the velocity vector of the UAV and $\varphi^{(t)}$ and $\varrho^{(t)}$ are respectively the azimuth and the elevation angles between the current UAV position and the desired destination. Note that, in general, $\boldsymbol{\nu}^{(t)}$ (2) is given by

$$\boldsymbol{\nu}^{(t)} = \boldsymbol{\omega}^{(t)} \cdot P_{vmax} \tag{3}$$

where P_{vmax} is the highest speed at which the UAV can fly. However, in the case where the UAV is closer than $\tau = 4$ meters to an intermediate waypoint, $\boldsymbol{\nu}^{(t)}$ (2) is calculated according to

$$\boldsymbol{\nu}^{(t)} = \boldsymbol{\eta}^{(t)} \cdot \left(\frac{\boldsymbol{\lambda}^{(t)}}{\boldsymbol{\tau}}\right)^{\gamma} \tag{4}$$

where $\boldsymbol{\eta}^{(t)} = \boldsymbol{x}_{\text{dest}} - \hat{\boldsymbol{x}}^{(t)}$, $\boldsymbol{\lambda}^{(t)} = \|\boldsymbol{\eta}^{(t)}\|$ denotes the distance between the current UAV's position estimate and the desired destination, $\boldsymbol{\tau}$ is a threshold that evaluates the proximity of the UAV to the current waypoint, while $\boldsymbol{\gamma}$ is a smoothing factor.

It becomes obvious from (1) that in order to govern the movement of the UAV, one first needs to determine its position. This can be done by exploiting the terrestrial radio measurements in the following way. Distance (range) measurements can be extracted from the received signal from TOF, RSS or some other property of the radio signal. In general, the k-th distance sample ($1 \leq k \leq K$) between the UAV at t and a reference point i can be modelled as

$$d_{i,k}^{(t)} = \|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\| + n_{i,k}^{(t)}, \;\; i = 1, ..., N, \;\; k = 1, ..., K, \tag{5}$$

where the notation $\| \bullet \|$ is used to represent the Euclidean norm and $n_{i,k}^{(t)}$ denotes the measurement noise, modelled as a zero-mean Gaussian random variable with standard deviation $\sigma_{i,k}^{(t)}$, i.e., $n_{i,k}^{(t)} \sim \mathcal{N}(0, \sigma_{i,k}^{(t)})$. Note that the median of the K distance samples, $d_i^{(t)}$, is used in the following derivation for the sake of simplicity and also to achieve greater robustness to outliers.

According to the maximum likelihood (ML) criteria [24] and from (5), the UAV's position at t could be determined according to

$$\widehat{\boldsymbol{x}}^{(t)} = \underset{\boldsymbol{x}^{(t)}}{\arg\min} \sum_{i=1}^{N} \frac{1}{2(\sigma_i^{(t)})^2} \left(d_i^{(t)} - \|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\|\right)^2. \tag{6}$$

However, the objective function in (6) is highly non-convex and, thus, might have multiple local optima, which aggravate addressing the ML estimator directly. Therefore, this work does not tackle (6) directly, but proposes a novel hybrid LSTM-based network to accomplish accurate UAV navigation. It makes use of the solutions of two existing model-based estimators during a portion of the trajectory of interest in order to train the network and adjust its parameters, after which it becomes sufficiently autonomous to accurately govern the movement of the UAV.

III. The Proposed LSTM Method

In this section, a detailed description for the proposed deep learning algorithm is provided. To better organize the work, this section is divided in two parts. The first sub-section explains the deep-neural network architecture, while the second sub-section explains how the network was trained.

A. Network Architecture

The network architecture is based on the use of LSTM, which is a type of Recurrent Neural Networks (RNN), trained applying back propagation through time.

The LSTM networks differ from other types of neural networks, since the network parameters are shared along the several points of the time sequence under analysis. Each one of the output elements is influenced by the outputs of some past elements. This particularity of persistence of the characteristics present in the data, allows the model to look at a sequence as

a whole, and not just at an individual element. Unlike other networks, RNNs use feedback from activations from previous time slots as input from the network to make a decision for the current input. It should be noted that the sequences that are analyzed by the RNN correspond to a time interval δ, which may not refer exactly to the phenomenon of time duration in the real world.

Using the RNN as a feature extraction layer it is necessary to use a Feedforward neural networks (FNN) to estimate the final position.

Therefore, the proposed network is composed of two LSTMs, designed in such a way that each of them corresponds to the outputs of a model-based estimator, from which they are trained.

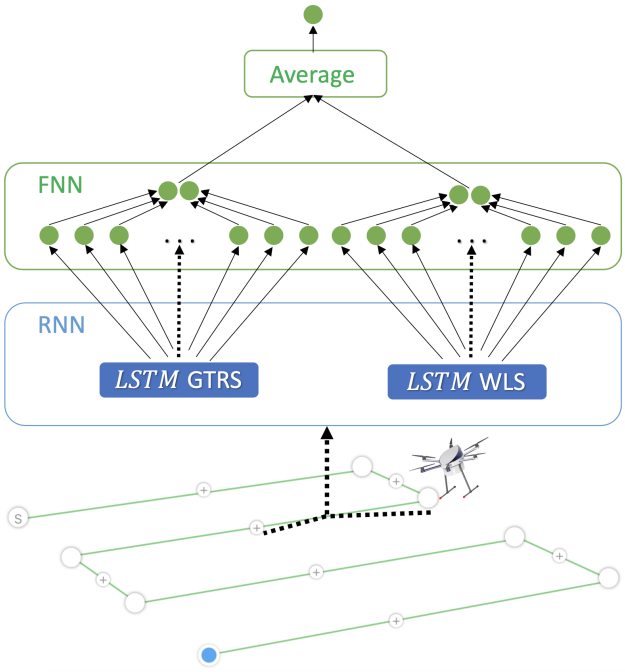


Fig. 1. The proposed deep-neural network.

These LSTMs of which the network is composed are fed by a dataset of the GTRS [20] and the WLS [21] estimations, respectively, which are averaged afterwards in order to further enhance the positioning accuracy. This is illustrated in Fig. 1.

It is important to mention that the output layer (each fully connected layer is activated by a ReLU:

$$\boldsymbol{F}^{(t)} = max(0, x) \tag{7}$$

Table I summarizes all layers and its hyperparameters for one neural network architecture, where N_n denotes the amount of neurons. It should be noted that GTRS and WLS use the same neural network architecture. The values present in the table were obtained using the summary function available from keras and tensorflow libraries.

A brief overlook of the statistical methods is provided in the following sub-sections.

TABLE I
THE PROPOSED RNN ARCHITECTURE HYPERPARAMETERS.

Layer	Output	Hyperparameters
INPUT	4	
LSTM	2	$N_n = 265216$;
FC OUTPUT	2	$N_n = 514$; optm=ADAM loss=mean_squared_error

B. Network Training

1) GTRS: The GTRS algorithm proposed in [20] overcomes the difficulties characteristic for the ML problem by transforming it into a GTRS optimization problem. This is accomplished via the introduction of the weights, $w_i^{(t)} = \frac{(d_i^{(t)})^{-1}}{\sum_{j=1}^{N}(d_i^{(t)})^{-1}}$, such that more trust is assigned to *closer* links. Afterwards, by applying a squared-range approach and developing the squared-term, it is possible to rewrite the problem in a vector form as

$$\begin{aligned} &\underset{\boldsymbol{z}^{(t)}}{\text{minimize}} \ \|\boldsymbol{A}^{(t)}\boldsymbol{z}^{(t)} - \boldsymbol{b}^{(t)}\|^2 \\ &\text{subject to} \\ &(\boldsymbol{z}^{(t)})^T\boldsymbol{D}\boldsymbol{z}^{(t)} + 2\boldsymbol{f}^T\boldsymbol{z}^{(t)} = 0, \end{aligned} \tag{8}$$

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

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

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

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

with $\hat{\boldsymbol{\theta}}^{(t|t-1)}$ and $\hat{\boldsymbol{P}}^{(t|t-1)}$ being respectively the mean and the covariance of the one-step predicted state (for more details, the reader is referred to [20]), while $\boldsymbol{I}_G$ and $\boldsymbol{0}_{p\times q}$ represent the identity matrix of size G and the matrix of all-zero entries of size $p \times q$.

The main attribute of a GTRS is that it there is a computable interval on which it is a monotonically decreasing function. Hence, it can be solved by simply applying the bisection procedure [25].

2) WLS: The WLS algorithm in [21] is a two-step scheme that uses geometry (equation of spheres) to get a coarse estimate of the UAV's position in order to fabricate azimuth and elevation angle estimates and *linearize* the positioning problem by applying Cartesian to spherical coordinate conversion. In order to accentuate the importance of *nearby* links it ponders them by introducing weights, $w_i^{(t)} = \frac{(d_i^{(t)})^{-1}}{\sum_{i=1}^{N}(d_i^{(t)})^{-1}}$. The WLS estimator [21] is given by

$$\underset{\boldsymbol{x}^{(t)}}{\text{minimize}} \|\boldsymbol{W}^{(t)}\left(\boldsymbol{H}^{(t)}\boldsymbol{x}^{(t)} - \boldsymbol{h}^{(t)}\right)\|^2 \tag{9}$$

where $\boldsymbol{W}^{(t)} = \text{diag}(\boldsymbol{w}^{(t)})$, $\boldsymbol{w}^{(t)} = \left[w_i^{(t)}\right]^T$,

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

and

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

$\boldsymbol{u}_i^{(t)} = [\cos(\hat{\phi_i}^{(t)})\sin(\hat{\alpha_i}^{(t)}), \sin(\hat{\phi_i}^{(t)})\sin(\hat{\alpha_i}^{(t)}), \cos(\hat{\alpha_i}^{(t)})]^T$ being the unit vector, and

$$\hat{\phi}_i^{(t)} = \arctan\left(\frac{\hat{x}_{i,2}^{(t)} - a_{i,2}}{\hat{x}_{i,1}^{(t)} - a_{i,1}}\right) \tag{10}$$

and

$$\hat{\alpha}_i^{(t)} = \arccos\left(\frac{\hat{x}_{i,3}^{(t)} - a_{i,3}}{\|\hat{\boldsymbol{x}}_i^{(t)} - \boldsymbol{a}_i\|}\right) \tag{11}$$

representing respectively the estimated azimuth and elevation angles from the coarse estimate of the UAV's position. Its final solution is given in closed-form as

$$\begin{aligned}\hat{\boldsymbol{x}}^{(t)} &= \left((\boldsymbol{H}^{(t)})^T(\boldsymbol{W}^{(t)})^T\boldsymbol{W}^{(t)}\boldsymbol{H}^{(t)}\right)^{-1} \\ &\times \left((\boldsymbol{H}^{(t)})^T(\boldsymbol{W}^{(t)})^T\boldsymbol{W}^{(t)}\boldsymbol{h}^{(t)}\right).\end{aligned} \tag{12}$$

3) UAV Navigation: Once the UAV position is estimated (either via GTRS or WLS, or some other method) and given that the desired (intermediate) destination of the UAV is known, the direction of the UAV can be computed by

$$\hat{\varphi}^{(t)} = \arctan\left(\frac{x_{\text{dest},2} - \hat{x}_2^{(t)}}{x_{\text{dest},1} - \hat{x}_1^{(t)}}\right), \tag{13a}$$

$$\hat{\varrho}^{(t)} = \arccos\left(\frac{x_{\text{dest},3} - \hat{x}_3^{(t)}}{\|\boldsymbol{x}_{\text{dest}} - \hat{\boldsymbol{x}}^{(t)}\|}\right), \tag{13b}$$

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

Therefore, the UAV navigation can be accomplished by

$$\hat{\boldsymbol{x}}^{(t+1)} = \boldsymbol{S}\left[(\hat{\boldsymbol{x}}^{(t)})^T, (\hat{\boldsymbol{v}}^{(t)})^T\right]^T, \tag{14}$$

where

$$\hat{\boldsymbol{v}}^{(t)} = \boldsymbol{\nu^{(t)}}\begin{bmatrix}\cos(\hat{\varphi}^{(t)})\sin(\hat{\varrho}^{(t)}) \\ \sin(\hat{\varphi}^{(t)})\sin(\hat{\varrho}^{(t)}) \\ \cos(\hat{\varrho}^{(t)})\end{bmatrix} \in \mathbb{R}^3 \tag{15}$$

is the estimated velocity vector at t. Note that, one is using an estimate of the UAV's position in (14), unlike (1).

To summarize this section, a pseudo-code of the proposed solution is presented in Algorithm 1.

Algorithm 1 The proposed LSTM algorithm

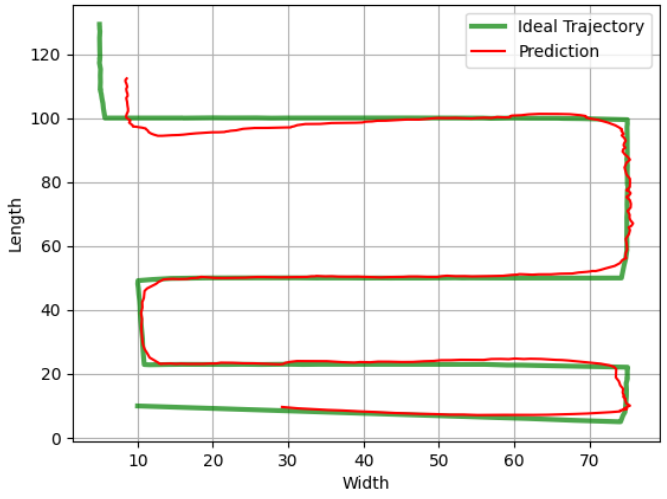
Initialization: $LB \leftarrow 10$, $TP \leftarrow 0.8$, $Eps \leftarrow 15$
Model definition: $ModelWLS$, $ModelGTRS$
`//Model training`
$trainGTRS \leftarrow$ Read GTRS file with $\sigma = 1$
$trainWLS \leftarrow$ Read GTRS file with $\sigma = 1$
`//Format dataset`
$XWLS$, $YWLS$ = createDataset()
$ModelWLS$.fit($XWLS$, $YWLS$)
$ModelGTRS$.fit($XGTRS$, $YGTRS$)
`//Save Model`
$ModelWLS$.save(), $ModelGTRS$.save()
`//Model prediction`
$testGTRS \leftarrow$ Model Predict for $\sigma = 3$
$testWLS \leftarrow$ Model Predict for $\sigma = 3$
$finalPrediction \leftarrow$ Average between testGTRS and testWLS for every position

IV. Performance Analysis

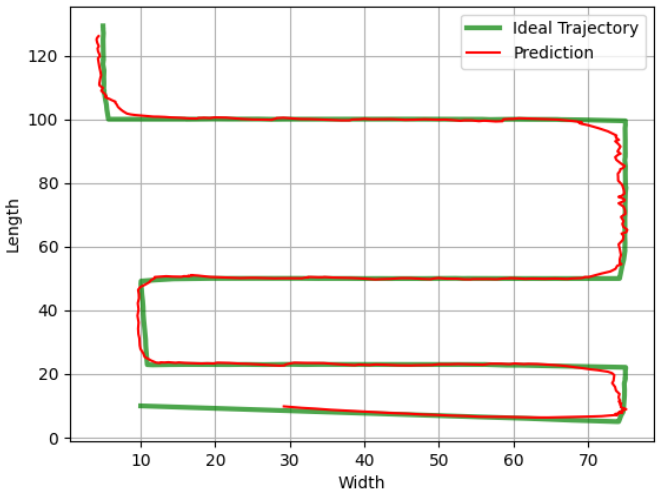
In this section, the performance of the proposed hybrid LSTM network is assessed by focusing on improvements against its counterparts, by considering GTRS-only and WLS-only algorithms. This is achieved through a sequence of tests performed utilizing Python programming language. Note that the datasets used to train the LSTM model were created by exploiting the GTRS [20] and the WLS [21] algorithms in MatLab® R2019b. The following parameters were defined in order to get a better comprehension of the following analysis. First, the number of previous time steps used to make a predictions for the subsequent time instant $LookBack(LB)$. Next, the number of $Epochs(Eps)$, defining how many times the learning model trains throughout the entire dataset. In all the simulations the value of Eps = 15 was considered. Lastly, the $TrainingPercentage(TP)$ which defines how many observations in a dataset are used in training the model. These parameters will be subject of analysis through simulation. If not stated differently, in the following simulations, the LSTM is trained using one dataset of GTRS predictions and one dataset of WLS predictions where $\sigma = 1$ meter is set. These datasets are then used to create a model which is used to make a prediction for $\sigma = 3$ meters.

First, the effect of the variation of TP is studied. Fig. 2 shows the obtained results. From this figure it is possible to conclude that, the predicted trajectory is closer to the ideal one when the whole dataset is used. This figure also indicates that the LSTM model does not need to use the complete dataset in the training stage to make relatively good predictions.

Afterwards, the variation of the LB parameter was studied, whose results are shown in Fig. 3. On the one hand, this figure shows that even with only using 5 previous time instants, the

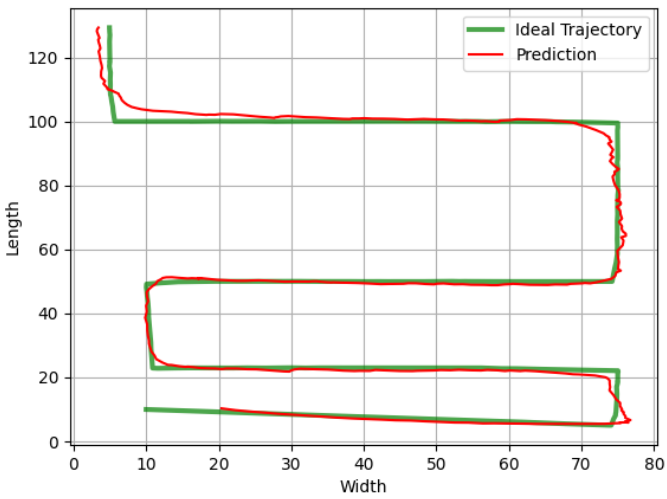


(a) TP = 80%

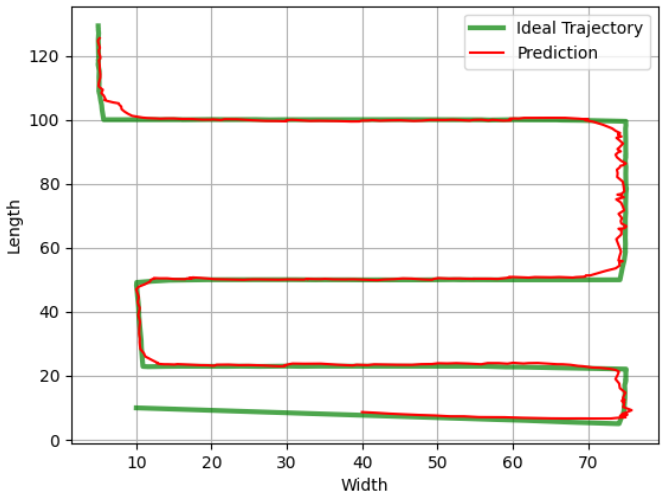


(b) TP = 100%

Fig. 2. Comparison between the desired trajectory of the UAV and the LSTM with LB = 10 and varying TP.



(a) LB = 5



(b) LB = 15

Fig. 3. Comparison between the desired trajectory of the UAV and the LSTM with TP = 100% and varying LB.

predictions made by the LSTM are relatively good. On the other hand, using 15 previous time instants a nearly perfect trajectory is achieved. The only noticeable differences between the ideal and the predicted trajectory are when the UAV needs to change its direction.

Finally, a comparison between the LSTM of the GTRS, the LSTM of the WLS and the combination of both algorithms is shown in Fig. 4. One can observe that, even though the two individual LSTMs achieve relatively good results, their combination into a hybrid LSTM results in a nearly perfect trajectory. This behavior is somewhat intuitive and corroborates our expectations that better results can be achieved by taking advantage of an increased quantity of the available information.

It is also crucial to compare the algorithms with the main performance metric: the average root mean square error (ARMSE), which is defined as

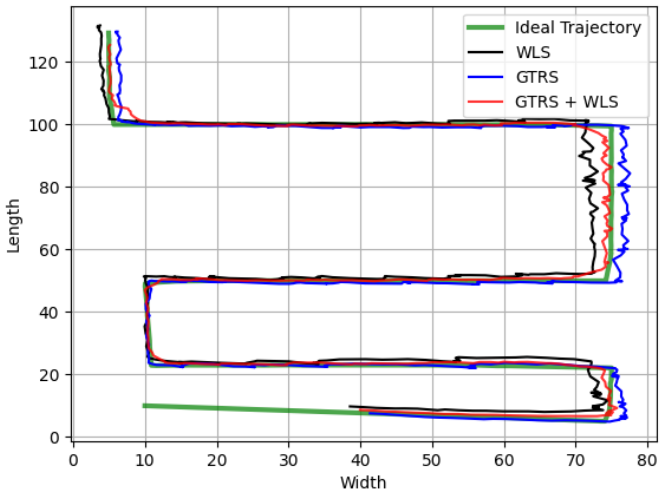


Fig. 4. Comparison between the WLS LSTM, the GTRS LSTM and the combination of GTRS and WLS with TP = 100%, LB = 15.

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

where $\boldsymbol{x}_{ij}$ and $\hat{\boldsymbol{x}}_{ij}$ respectively denote the true and the estimated UAV's location in the i-th Monte Carlo (Mc) run and the j-th mission waypoint (Mw).

Table II shows the obtained results for the GTRS LSTM, WLS LSTM and the combination of both predictions using the average between them.

TABLE II
ARMSE (M) FOR DIFFERENT PARAMETERS AND TIME CONSUMPTION COMPARISON.

Parameters		ARMSE (m)		
		GTRS in [20]	WLS in [21]	**The proposed work**
$N = 8$	$\sigma_i^{(t)} = 1$	3.76	3.73	3.31
	$\sigma_i^{(t)} = 2$	3.80	3.89	3.38
	$\sigma_i^{(t)} = 3$	3.86	3.88	3.52

Table II, shows that the proposed algorithm achieves a lower ARMSE when compared with the two separate LSTMs. This can be explained due the quantity of information available. Therefore, it is possible to conclude that the proposed work is somewhat more robust to noise. Note that, even the smallest deviation from the ideal trajectory can result in a collision with an obstacle.

V. Conclusion and future work

In this work, a new deep-neural network which is able to improve current solutions that solve the problem of UAV navigation in satellite-less environments was proposed. Firstly, the GTRS and WLS predictions are used to create two datasets. These datasets are then used to train the network composed of two LSTMs, one for each algorithm. Secondly, the mean of the two LSTM predictions is combined and a final estimate of the position is made. The results show that not only did the navigation accuracy improve, but also did the robustness to noise. Note that this is an important result since, in indoor environments for instance, a slight deviation from the ideal path can result in a collision with an object. In this way, it is possible to say that the presented algorithm can be used and adapted to existing methods to further enhance their accuracy.

Future work includes the study of other types of RNNs, such as Convolutional Neural Networks, to set a new bound on the achievable navigation accuracy and the inclusion of obstacle avoidance in order to achieve a fully autonomous navigation (without the use of intermediate points).

Acknowledgment

This research was partially funded by by the European Union's Horizon Europe Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement No. 101086387, and Fundação para a Ciência e a Tecnologia under Projects UIDB/04111/2020, UIDB/50008/2020, ROBUST EXPL/EEI-EEE/0776/2021, and 2021.04180.CEECIND, as well as Instituto Lusófono de Investigação e Desenvolvimento (ILIND) under Project COFAC/ILIND/COPELABS/1/2022.

References

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

[2] S. D. 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.

[3] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, "An autonomous multi-uav system for search and rescue," in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ser. DroNet '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 33–38. [Online]. Available: https://doi.org/10.1145/2750675.2750683

[4] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[5] S. K. Khan, U. Naseem, H. Siraj, I. Razzak, and M. Imran, "The role of unmanned aerial vehicles and mmwave in 5g: Recent advances and challenges," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, p. e4241, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4241

[6] S. Sulemane, J. P. Matos-Carvalho, D. Pedro, F. Moutinho, and S. D. Correia, "Vineyard gap detection by convolutional neural networks fed by multi-spectral images," *Algorithms*, vol. 15, no. 12, 2022. [Online]. Available: https://www.mdpi.com/1999-4893/15/12/440

[7] G. Mestre, J. P. Matos-Carvalho, and R. M. Tavares, "Irrigation management system using artificial intelligence algorithms," in *2022 International Young Engineers Forum (YEF-ECE)*, 2022, pp. 69–74.

[8] A. B. Salvado, R. Mendonça, A. Lourenço, F. Marques, J. P. Matos-Carvalho, L. Miguel Campos, and J. Barata, "Semantic navigation mapping from aerial multispectral imagery," in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, 2019, pp. 1192–1197.

[9] E. Innocenti, G. Agostini, and R. Giuliano, "Uavs for medicine delivery in a smart city using fiducial markers," *Information*, vol. 13, no. 10, 2022. [Online]. Available: https://www.mdpi.com/2078-2489/13/10/501

[10] Z. Zhao, P. Cumino, C. Esposito, M. Xiao, D. Rosário, T. Braun, E. Cerqueira, and S. Sargento, "Smart unmanned aerial vehicles as base stations placement to improve the mobile network operations," *Computer Communications*, vol. 181, pp. 45–57, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366421003534

[11] K. Máthé and L. Buşoniu, "Vision and control for uavs: A survey of general methods and of inexpensive platforms for infrastructure inspection," *Sensors*, vol. 15, no. 7, pp. 14 887–14 916, 2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/7/14887

[12] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2, 2009, pp. 82–85.

[13] 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.

[14] 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.

[15] 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.

[16] J. P. 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, p. 2501, Oct 2019. [Online]. Available: http://dx.doi.org/10.3390/rs11212501

[17] J. Nakama, R. Parada, J. P. Matos-Carvalho, F. Azevedo, D. Pedro, and L. Campos, "Autonomous environment generator for uav-based simulation," *Applied Sciences*, vol. 11, no. 5, p. 2185, Mar 2021. [Online]. Available: http://dx.doi.org/10.3390/app11052185

[18] D. Pedro, A. Mora, J. Carvalho, F. Azevedo, and J. Fonseca, "Colanet: A uav collision avoidance dataset," in *Technological Innovation for Life Improvement*, L. M. Camarinha-Matos, N. Farhadi, F. Lopes, and H. Pereira, Eds. Cham: Springer International Publishing, 2020, pp. 53–62.

[19] 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.

[20] J. P. 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.

[21] 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, pp. 93–102, June 2022.

[22] B. Yang, J. Li, Z. Shao, and H. Zhang, "Robust UWB indoor localization for NLOS scenes via learning spatial-temporal features," *to appear in IEEE Sensors Journal*, vol. XX, no. XX, pp. 1–1, Mar. 2022.

[23] 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.

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

[25] S. Tomic, M. Beko, and R. Dinis, "3-D target localization in wireless sensor network using RSS and AoA measurement," *IEEE Trans. Vehic. Technol.*, vol. 66, no. 4, pp. 3197–3210, Apr. 2017.