Wireless Signal Source Localization by Unmanned Aerial Vehicle using AERPAW Digital Twin and Testbed

BAISAKHI CHATTERJEE, SONALI CHAUDHARI, ZHIZHEN LI, YUCHEN LIU, RUDRA DUTTA
Department of Computer Science, North Carolina State University
{*bchatte, sschaud2, zli92, yliu322, rdutta*}*@ncsu.edu*

***Abstract*—Unmanned Aerial Vehicles (UAVs) have a variety of applications, making them a focus of growing research interest. Estimates say that there could be over three million aerial vehicles taking to the skies in the next few years. Thus, it has become critical to develop and test mission-specific algorithms without compromising safety. Currently, FAA rules allow only Line-of-Sight (LoS) testing for Aerial Vehicles, which limits the number of applications that can be verified. Furthermore, validating infrastructure-intensive algorithms in the field is not feasible for every researcher. To mitigate this issue, remote research infrastructure testbeds have been developed. AERPAW is one such project funded by US NSF where researchers can develop their programs in a Digital Twin simulation environment which the AERPAW team will then test. In this paper, we have utilized the AERPAW testbed to develop a localization algorithm for finding the source of a radio signal using a single UAV. We observe the results and note the differences between the emulation and real-world conditions.**

***Index Terms*—Unmanned Aerial Vehicle, autonomous agents, source localization, testbed, drone, aerial wireless network**

I. INTRODUCTION

In recent times, Unmanned Aerial Vehicles have gained significant popularity owing to their maneuverability and versatility of applications. While UAVs started as exclusively military tools [1], more and more of them are now being deployed for civilian use. Autonomous UAVs, in particular, have been utilized for a variety of applications, such as, search and rescue [2], target monitoring [3] and disaster surveillance [4]. With millions of UAVs set to take to the air in the next few years [5], it has become increasingly important to not only theoretically develop and test algorithms for UAV functionality but also verify correctness and safety in the real world. However, it is not always feasible for researchers to build and deploy the necessary infrastructure. The Aerial Experimentation and Research Platform for Advanced Wireless (AERPAW) was developed to solve this issue.

AERPAW is a national aerial wireless experimentation platform that provides remote access to researchers for flexible and programmable experiments focused on wireless research and unmanned vehicles. AERPAW offers adaptability and resources to researchers, allowing them to develop their algorithms in an emulation environment, which is the Digital Twin [6] of ground conditions, and then, see the outcome of their theories in the real world. However, uncertainty in the real world means that not all aspects line up completely. In this paper, we demonstrate the differences observed when an experiment is run in the AERPAW emulation environment versus the actual testbed. To do this, we have implemented a Source Localization algorithm in AERPAW where a UGV hidden in a field emits a radio signal and a UAV must find the source of the signal. Our results show that, in case of AERPAW, while the real world implementation is very close to the emulation environment, some key differences exist. This experiment was conducted as part of the AERPAW Find A Rover (AFAR) Challenge. We note that this manuscript includes information including images from the AERPAW website [7].

This research was supported by the National Science Foundation through the PAWR Program Grant No. CNS–1939334, its supplement for studying NRDZ and through the National Science Foundation Award CNS–2312138.

ISBN 978-3-903176-63-8

II. Relevant Prior Work

As stated previously, our goal in this paper was to observe and analyze how real world conditions could add to uncertainties which were not present in the emulation environment. As such, the choice of our experiment, though constrained by the challenge requirements, was still uniquely suited as an exemplar problem for our task. Localization is the problem of determining the position of an object by measuring some emission like odor, sound, or radio waves. It has a variety of applications and is rapidly becoming an area of considerable interest. Problems can consist of finding the source of a pollutant [8] or gaseous leak [9], to locating a sound source [10] or mapping a node in a wireless sensor network [11]. Fink and Kumar [12] in their paper, presented an online Gaussian model for approximating the location of a static source by measuring the signal strength of a radio in an indoor setting. Lee et al. [13] developed a cost-effective signal measurement model for indoor localization using fingerprinting technology. Fan et al. [14] presented a method for 3D indoor localization using a Structured Bidirectional Long Short-term Memory (SBi-LSTM) architecture.

While outdoor localization problems have usually focused GPS-based vehicular tracking, in some cases, GPS might not be available and other methods may be necessary. Bhattacharjee et al. [15] presented a tracking algorithm for detecting a UAV using only passive RF signals. Al-Sadoon et al. [16] tackled the issue of asset tracking in a dense GPS-unfriendly environment by utilizing a compact-size sensor array of six electrically small dual-band omnidirectional spiral antenna elements for the angle of arrival (AOA) method. Kwon and Guvenc [17] showed five methods for estimating a static RF signal source using a linear least square (LLS) based localization scheme and a waypoint-based UAV trajectory.

III. Problem Model

The goal of our study is to implement and validate outdoor source localization under real-world circumstance. The source signal is emitted by a stationary Unmanned Ground Vehicle (UGV) which is located somewhere in the area defined by the green square, as can be seen in Fig. 1. The UGV has one transmit antenna and continuously emits a GNU Radio-based channel-sounding narrowband waveform with a bandwidth of 125KHz. A UAV will be able to receive this signal using software-defined radios and determine the strength of the signal based on its distance to the source. Three tests were conducted, two with the rover hidden in the area the UAV is allowed to fly over and one outside this region. This allowed us to test for scenarios where a signal source may be located in a restricted or dangerous zone, e.g., somewhere with extreme wind conditions [18], in addition to regular cases where the UAV can search an entire region. Two difference testing scenarios were selected to judge the accuracy of the algorithm. The first scenario consisted of a fast estimation and allowed the UAV to fly for a total of 3 minutes before reporting its findings. For a more robust estimation, allowed the UAV to take a longer time (10 min) so that more samples could be collected.

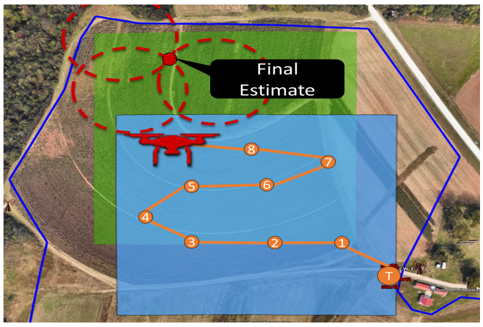


Fig. 1: Illustration of UAV traversing the field using waypoints. Based on the estimated signal strength, the UAV calculates the circular area the source could be in. The intersection of 3 such circles provides the final estimate.

A. AERPAW Architecture

AERPAW is a batch-mode facility consisting of a set of virtual computing nodes, each aligning with a physical AERPAW computing node. The programs can be tested in a simulation environment which is the Digital Twin of real-world conditions. To ensure an easy learning curve, the *AERPAW software* provides a range of extendable modules that allow the user to easily program each node. In this section, we will briefly describe the necessary parts required in this experiment.

Software Libraries

The AERPAW software library can be roughly separated into two categories - *Programmable Entities* and *Backplane*. The programmable entities are those exposed to the user and can be modified as required. These include vehicle control software, GNU Radio, and some utility functions. The vehicle control software is designed as a State Machine where users can define their states. A single thread keeps track of which state the vehicle is in, while another may interrupt the current thread to enforce some action. The *aerpawlib* package provides functions that allow users to convert GPS coordinates to angle headings and vice versa. To communicate with the drones, we use MAVLink. The secure Backplane is responsible for isolating the platform with the rest of the world and the experiments from each other.

Hardware Nodes

AERPAW provides several resources that the user can program as per their requirements. A computing system, commonly denoted as a Companion Computer, is equipped with client-side AERPAW control software and accompanies the experimental resources (such as radios or vehicles) which researchers aim to investigate. This is referred to as an AERPAW Node. When physically instantiated for an experiment, it becomes an AERPAW Hardware Node (AHN). Nodes are of two types - fixed and portable. In this experiment, we focus only on the portable nodes, which are The Large AERPAW Multirotor (LAM) and the Unmanned Ground Rover. All AERPAW nodes use USRP radio software to communicate with the central system.

IV. Our Approach

To localize the source, we adopted the Linear Least Square method to estimate the position of the rover based on repeated sampling of the received signal. Since the experiment would be conducted on an open field, we have considered that the UAV will maintain a Line of Sight (LoS) connectivity with the signal source and, thus, have chosen the free space path loss model. The formula for calculating path loss is given by [19]

$$P_d = P_0 + 10\alpha \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma,$$

where P_0 is the path loss at some reference distance d_0 and X_σ is a Gaussian random variable with mean 0 and standard deviation σ. α refers to the path loss coefficient. d refers to the distance between the ground target and the UAV, $d_i = f(x_i, y_i, z_i) = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}$ where (x, y, z) is the real target location. For the reference point we have used field test data provided by the AERPAW team.

Positioning in AERPAW follows the NED (North, East, Down) system. To move the vehicle, users can provide GPS coordinates which converted internall by the software library to relative directions of angle and distance. Alternatively, users can explicitly turn the vehicle in the direction required and move along a specified vector. The movement command is sent to the MAVLink, which executes the actual displacement of the vehicle. Error tolerance by defaults is $2m$. The UAV will take off from the bottom right, as shown in Fig. 1, marked in red. While the UAV is allowed to fly in the entire region marked by the blue square in this experiment, it is evident that the rover cannot be placed in the region which comprises of only the blue square but not the green one. Thus, in our algorithm, the UAV flies to the bottom right corner of the green square to begin its search. Once the UAV reaches the corner, it follows a set of predefined waypoints present inside the blue square region and measures the power at each point. A safety checker has been constructed to ensure that the UAV remains within the boundaries of the defined geofence. While the dark blue line, as shown in Fig. 1, represents the general UAV boundaries for all AERPAW experiments, the blue square is the flight boundary for the UAV. Before moving to a waypoint, the UAV checks whether the coordinate is within the boundary and begins flight if it is so. If not, the UAV moves on to the next waypoint. Once the list of waypoints has been exhausted, it keeps moving forward and measures power after every $40m$. If, at any point, it hits the geofence, the UAV turns right and repeats the process.

At each location, the UAV hovers till a predermined (5 in our case) number of measurements are recorded. This ensures that random noise does not distort the signal power. We also take into account the quality of the signal recorded. Power and Quality are measured as per GNURadio parameters with frequency $3.32GHz$. The UAV then records the weighted measurement of power as

$$P_i = \frac{1}{m}\sum_{j}^{m=5} P_i^j * Q_i^j$$

where P_i^j and Q_i^j are the j^{th} measurement of power and quality respectively at the i^{th} waypoint. P_i is the weighted i^{th} power measured. Power is measured in decibels, while quality is measured in decimals, with 1 being the highest. Once the measurements are done, the UAV moves on to the next waypoint.

Determining the location of the ground target involves obtaining distance measurements to the target from at least three distinct UAV anchor positions. In addition to this, for any LLS-based localization approach, the r-th index must be selected as a reference point. In our experiment, we have set $r = 0$, which means the first point is selected as the reference. Once at least 4 power measurements have been completed, the UAV calculates the estimated target location as

$$l = \frac{A_v^T b_v}{A_v^T A_v}$$

where A_v and b_v are computed as in [17]. All data obtained are stored in logs maintained by the UAV, which the experimenter can obtain and evaluate. For this experiment, we stored power and quality measurements, as well as the best estimate of the rover at different points. As mentioned previously, the best estimates at 3min and 10min are used for evaluation criteria. The experiment is concluded after the assigned 10 minutes are over, at which point the UAV returns to its home coordinates. Details of our algorithm can be found in Algorithm 1.

Algorithm 1 : Rover Search Approach

Procedure: *search (vehicle* : *Drone)*

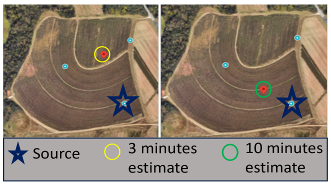
await $TakeOff$
while $time_{now} < time_{allowed}$ **do**
    await goTo($getBottomRightCoords()$)
    $msr_{all}.append(sum(power * quality)/m)$
    if $msr_{all}.size() > 4$ **then**
        $ref\_pts = msr_{all}.pop(0)$
        $x_r, y_r, z_r = ref\_pts[0]$
        $d_r = ref\_pts[1]$
        $xxx = x_r^2 * \alpha + y_r^2 * \alpha * 0.64$
        for $i \in range(1, msr_{all}.size())$ **do**
            $x_{val} = msr_{all}[i][0].lat - x_r * \alpha * 2$
            $y_{val} = msr_{all}[i][0].lon - y_r * \alpha * 1.28$
            $b_{val} \leftarrow d_r^2 - msr_{all}[i][1]^2$
            $A.append([x_{val}, y_{val}]$
            $b.append(b_{val})$
        $A^T \leftarrow transpose(A)$
        $l \leftarrow (A^T * b)/(A^T * A)$
        $best\_position \leftarrow (l.lat, l.lon, z_r)$
    $moveCoord \leftarrow waypointList.pop()$
    while $isOutsideFence(moveCoord)$ **do**
        if $waypointList.size() > 0$ **then**
            $moveCoord \leftarrow waypointList.pop()$
        else
            $moveCoord \leftarrow turnRight()$
    await goTo($moveCoord$)
await goTo($homeCoords$)
await $Land$

V. Evaluation Results

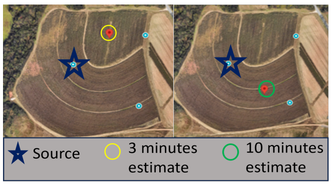
As mentioned previously, AERPAW is a batch-mode facility, which means that every program is tested virtually before being executed in the field. For this challenge, both virtual and physical testing was conducted by the AERPAW team. Once the virtual testing satisfied safety considerations, only then were UAVs tested in the actual testbed. For both virtual and physical (testbed) testing, the algorithm is evaluated three times. Each time, UGV is kept at a different known location, as shown in Fig. 2. The AERPAW team ran the algorithm to find the 3 min speed estimation and 10 min accuracy estimation for each test. The distance between the estimated location and the actual UGV location called an error, is calculated for each test. The error for three tests is then averaged to evaluate the algorithm's accuracy.

A. Emulation Testing

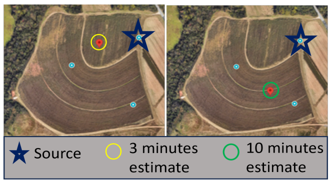
Emulation Testing is the first stage of verification conducted in the AERPAW testbed. AERPAW comprises two primary virtual entities: the E-VM and C-VM. The E-VM serves as the experimenter's virtual computer, granting root access. The experimental SDRs, along with the UAV/UGV (specifically its autopilot), are directly reachable from the E-VM



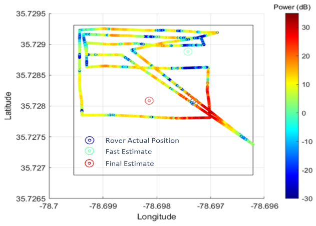
(a) Emulation - Test 1



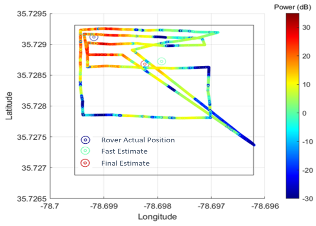
(b) Emulation - Test 2



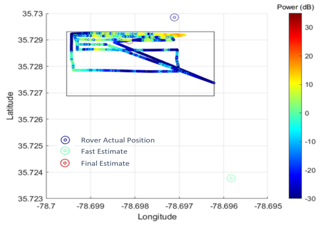
(c) Emulation - Test 3



(d) Testbed - Test 1



(e) Testbed - Test 2



(f) Testbed - Test 3

Fig. 2: Tests performed in Virtual Environment (Top) and AERPAW Test Bed (Bottom).

through APIs supplied by standard programming libraries. Additionally, each AERPAW Node runs a C-VM container, tasked with monitoring and providing override capabilities. This allows the AERPAW team to evaluate safety conditions, as well as FAA guideline compliance, and abort experiments which may violate regulations or lead to dangerous situations.

As per the evaluation criteria, location estimates were recorded for 3 and 10 minute marks, with the caveat that the experiment would be aborted if the UAV tried violate the geofence. Fig. 2 (Top) shows the test scenarios and results for our algorithm, containing actual position of the UGV, the 3 min estimate, and the 10 min estimate given by the algorithm in 3 tests. Table 1 shows the error in localization in all three instances (i.e. distance between source and estimation). The average error for the 3 min estimate was found to be 149.6m, and for 10 min was 142.86m.

B. *Testbed Testing*

As with the emulation testing, the final testing scenario also consisted of 3 runs with the UGV being placed in the same location as in the digital environment. Fig. 2 (Bottom) shows the actual position, the 3 min estimate, and the 10 min estimate by the algorithm for each test performed, while Table 1 shows the error in localization for these tests. The algorithm estimated the location of UGV with the error of 310m in 3 min and 119.4m in 10 min.

In both Emulation and Testbed scenarios, we observe that our estimations are better for the 10 min scenario, which is in line with our expectations that more number of samples will produce greater accuracy. However, it is interesting to note that each individual run does not provide better accuracy for the 10 minute results (Emulation Run 1). This oddity was not observed in the Testbed results which is likely because of using a weighted measurement of signal that helped get a better average. In an emulated environment, signal strength was not a strong indicator of power but in real life situations

TABLE 1. Emulation and Testbed Results

Category	***Time (min)***	***Run No***	***Error (m)***	***Average Error (m)***
Emulation	3	1	92.4	
		2	152.7	
		3	203.7	149.6
	10	1	155.4	
		2	134.1	
		3	139.1	142.9
Testbed	3	1	91.6	
		2	146.7	
		3	691.7	310.0
	10	1	93.1	
		2	116.2	
		3	148.8	119.4

signal power as well as quality were both necessary to correctly estimate signal strength. Additionally, we observed that in Testbed Run 3, our algorithm provided poor results for the fast (3 min) estimate when the rover was placed outside the UAV's flight zone, while this was not the case in the Emulated environment, indicating that processing of signal power in real-world scenario needs additional care.

VI. Conclusion & Future Work

In this paper, we have demonstrated our implementation of a wireless source localization algorithm using a UAV on the AERPAW testbed and compared its performance with the same algorithm on the Digital Twin Emulation environment. This algorithm was developed as a part of the AERPAW Find A Rover Challenge and, as such, we have discussed the limitations of our implementation based on the guidelines provided. Our results show that the AERPAW Digital Twin is a near-identical copy of the real life environments and can be depended upon by researchers to conduct their experiments safely. However, some changes do exist and real-life conditions can alter certain results.

We plan to perform further experiments on the testbed and are currently working on implementations where multiple UAVs need to communicate with each other in order to complete a task. Initially, we plan to do this by using the Backbone Message Queue channel and, in the future, extend this to utilize purely wireless communication.

References

[1] J. W. Canan, "Unmanned aerial vehicles," *Air Force Magazine*, 1988.

[2] J. Cho, J. Sung, J. Yoon, and H. Lee, "Towards persistent surveillance and reconnaissance using a connected swarm of multiple UAVs," *IEEE Access*, vol. 8, pp. 157906–157917, 2020. Conference Name: IEEE Access.

[3] T. Z. Muslimov and R. A. Munasypov, "Multi-UAV cooperative target tracking via consensus-based guidance vector fields and fuzzy MRAC," *Aircraft Engineering and Aerospace Technology*, vol. 93, no. 7, pp. 1204–1212, 2021. Publisher: Emerald Publishing Limited.

[4] A. Khan, S. Gupta, and S. K. Gupta, "Emerging UAV technology for disaster detection, mitigation, response, and preparedness," *Journal of Field Robotics*, vol. 39, no. 6, pp. 905–955, 2022.

[5] "UTM concept of operations version 2.0 (UTM ConOps v2.0)," Federal Aviation Administration, 2020. URL: https://www.faa.gov/sites/faa.gov/files/2022-08/UTM_ConOps_v2.pdf. Accessed: 29 Mar 23.

[6] T. Samal, R. Dutta, I. Guvenc, M. L. Sichitiu, B. Floyd, and T. Zajkowski, "Automating Operator Oversight in an Autonomous, Regulated, Safety-Critical Research Facility," in *2022 International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–10, July 2022. ISSN: 2637-9430.

[7] "Aerpaw • aerial experimentation and research platform for advanced wireless." URL: https://aerpaw.org/. Accessed: 22 Dec 23.

[8] Z. Fu, Y. Chen, Y. Ding, and D. He, "Pollution Source Localization Based on Multi-UAV Cooperative Communication," *IEEE Access*, vol. 7, pp. 29304–29312, 2019. Conference Name: IEEE Access.

[9] G. Bonow and A. Kroll, "Gas leak localization in industrial environments using a tdlas-based remote gas sensor and autonomous mobile robot with the tri-max method," in *2013 IEEE International Conference on Robotics and Automation*, pp. 987–992, 2013.

[10] J.-M. Valin, F. Michaud, J. Rouat, and D. Letourneau, "Robust sound source localization using a microphone array on a mobile robot," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 2, pp. 1228–1233 vol.2, 2003.

[11] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, 2007.

[12] J. Fink and V. Kumar, "Online methods for radio signal mapping with mobile robots," in *2010 IEEE International Conference on Robotics and Automation*, pp. 1940–1945, 2010.

[13] J. H. Lee, T. Kim, B. Shin, C. Yu, H. Kyung, and T. Lee, "Rf signal strength modeling in indoor environments for cost-effective deployment of fingerprinting technology," in *2022 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–4, 2022.

[14] S. Fan, Y. Wu, C. Han, and X. Wang, "A Structured Bidirectional LSTM Deep Learning Method For 3D Terahertz Indoor Localization," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pp. 2381–2390, July 2020. ISSN: 2641-9874.

[15] U. Bhattacherjee, E. Ozturk, O. Ozdemir, I. Guvenc, M. L. Sichitiu, and H. Dai, "Experimental Study of Outdoor UAV Localization and Tracking using Passive RF Sensing," in *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization*, WiNTECH '21, (New York, NY, USA), pp. 31–38, Association for Computing Machinery, Oct. 2021.

[16] M. A. G. Al-Sadoon, R. Asif, Y. I. A. Al-Yasir, R. A. Abd-Alhameed, and P. S. Excell, "Aoa localization for vehicle-tracking systems using a dual-band sensor array," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 8, pp. 6330–6345, 2020.

[17] H. Kwon and I. Guvenc, "RF Signal Source Search and Localization Using an Autonomous UAV with Predefined Waypoints," Jan. 2023. arXiv:2301.07027 [eess].

[18] B. H. Wang, D. B. Wang, Z. A. Ali, B. T. Ting, and H. Wang, "An overview of various kinds of wind effects on unmanned aerial vehicle," *Measurement and Control*, vol. 52, no. 7-8, pp. 731–739, 2019.

[19] Y. Zeng, I. Guvenc, R. Zhang, G. Geraci, and D. W. Matolak, eds., *UAV communications for 5G and beyond*. IEEE Press, Standards Information Network, Dec. 2020.