Non-myopic Sensor Path Planning for Emitter Localization with a UAV

Folker Hoffmann[1], Hans Schily[1], Markus Krestel[2], Alexander Charlish[1], Matthew Ritchie[3], Hugh Griffiths[3]

Abstract— **This paper addresses the problem of localizing a stationary RF emitter with a mobile UAV, equipped with a single directional antenna. By rotating around its vertical axis, it determines a bearing towards the emitter. Our interest is in optimally selecting the measurement positions to achieve a fast localization. The majority of such systems described in the literature use greedy planning to select the next measurement position. This work experimentally tests an algorithm that performs a non-myopic planning until the final localization step. The algorithm is based on the policy rollout principle and showed good performance in previous simulative studies. It is adapted to match the needs of a real world setup and evaluated in flight trials. Adaptions include the avoidance of close range measurements to prevent inaccurate measurements at high elevation, and the filtering of poor measurements.**

I. Introduction

Emitter localization is the problem of finding the position of a radio frequency (RF) emitter. It has applications in tracking radio-tagged animals, search and rescue, localization of RF interferences, and military surveillance. A localization requires measurements from multiple positions, motivating the use of mobile systems like unmanned aerial vehicles (UAVs). In this paper we consider a UAV equipped with a directional antenna. By performing a rotation around its vertical axis, it can measure the received power over the swept angle and compute a bearing towards the emitter. The quality of a localization with bearing only measurements is known to be geometry dependent. Different measurement positions lead to a faster or slower localization. Our goal is to perform measurements in a way that minimizes the time until the emitter is localized. For this, we developed a control algorithm that successively determines the next measurement position of the UAV.

The localization of RF emitters from mobile systems has attracted considerable interest in the sensor fusion and robotics literature. These systems can in general be categorized into those measuring the signal strength (received signal strength indicator, RSSI) and those that compute a bearing towards the emitter. RSSI-based systems [1]–[3] use the fading of signal power over distance to compute a range estimate for the emitter. Bearing-based systems compute a direction estimate towards the emitter, often by rotation of a directional antenna [3]–[10]. Using a combination of directional and omnidirectional antennas, an ambiguous pseudo-bearing can be computed without rotation [11].

Several of those systems implement automatic planning of the sensor path to maximize the system performance, often based on a greedy planner. [2] selects the next heading that maximizes the Rényi divergence over the next 5 s. In [1] a fast heuristic planner is presented which reduces the distance towards the emitter. Similarly, [9] moves towards the estimated source direction. [7] moves to a position on the minor axis of the uncertainty, with an adaptive distance. In [3] a formation of UAVs moves its formation center to the estimated target position. [4] selects the next measurement position, which minimizes the expected entropy of the next position estimate. A greedy planner is also used in [11]. The work in [8] assumes the target to be at the outer bound of the uncertainty and then selects the theoretically optimal two next measurement positions to localize a target on this position.

These systems mostly work in a myopic way, only considering a limited time horizon and often ignoring the uncertain outcomes of future measurements. In comparison, this paper presents a planning algorithm that plans from the current state until the end of the localization process, considering the full uncertainty of the current emitter position estimate and different actions in the future based on different outcomes of measurements. The control algorithm presented here was developed in previous work [12]–[14] and shows good performance in simulations. This paper is based on the presentation in [15] and describes the transfer of the algorithm to a real system and experimental results.

Section II formally states the planning problem. Section III presents the path planning algorithm and describes the adaptations necessary for experimental evaluation of the algorithm. The system and the experimental area are described in Section IV. We discuss the localization performance and the quality of our time prediction model in Section V. Finally, Section VI concludes the paper.

II. Problem Description

The state consists of the position of the emitting target $\mathbf{x}^t = (x^t, y^t)$ and the platform position $\mathbf{x}^p_k = (x^p_k, y^p_k)$, where k denotes the decision step. We model the joint state as the 4-dimensional vector

$$\mathbf{x}_k = (\mathbf{x}^t, \mathbf{x}^p_k) = (x^t, y^t, x^p_k, y^p_k) \in \mathbb{R}^4 \ . \qquad (1)$$

At each step, the system performs a measurement

$$z_k = \text{atan2}(y^t - y^p_k, x^t - x^p_k) + \mathcal{N}(0, \sigma^2) \qquad (2)$$

[1] Folker Hoffmann, Hans Schily, and Alexander Charlish are with the Sensor Data & Information Fusion department at the Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, Wachtberg, Germany. {firstname.secondname}@fkie.fraunhofer.de

[2] Markus Krestel is with the Schönhofer Sales and Engineering GmbH, Siegburg, Germany. At the time of this work, he was working at Fraunhofer FKIE.

[3] Matthew Richie and Hugh Griffiths are with the Department of Electrical Engineering, University College London, London, United Kingdom. {m.ritchie, h.griffiths}@ucl.ac.uk

of the angle between target and platform with additive Gaussian noise. We assume the target to be the only emitter in reception range and the standard deviation σ of the noise to be known. The duration of a measurement is modeled as a constant time t_M, during which the UAV must rotate without changing position.

After taking a measurement, the control algorithm selects the next sensing action. An action $\mathbf{a}_k \in \mathcal{A}_k \subseteq \mathbb{R}^2$ taken by the system represents the next measurement position. The UAV moves to this position, which is modeled with the transition function

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{a}_k) = (\mathbf{x}^t, \mathbf{a}_k) \ . \tag{3}$$

This movement takes an amount of time

$$t_D(\mathbf{x}^p_k, \mathbf{a}_k) = \frac{\|\mathbf{x}^p_k - \mathbf{a}_k\|_2}{v_p} \tag{4}$$

where v_p models the speed of the platform. Sensing and moving therefore lead to the joint cost of

$$c(\mathbf{x}^p_k, \mathbf{a}_k) = t_D(\mathbf{x}^p_k, \mathbf{a}_k) + t_M \ . \tag{5}$$

The emitter position is not observable from a single bearing measurement. Therefore, we integrate all measurements into a belief

$$b^t_k(\check{\mathbf{x}}^t) = p(\mathbf{x}^t = \check{\mathbf{x}}^t \mid b^t_0, \mathbf{a}_{0:k-1}, z_{1:k}) \tag{6}$$

which is a probability distribution over the emitter position. Here b^t_0 denotes prior knowledge. In our experiments this consists of the extension of the experimental area, on which it is uniformly distributed. The action $\mathbf{a}_0$ might either be the result of planning on the prior belief, or a given first measurement position. In the experiments $\mathbf{a}_0$ is explicitly specified. The belief about the full state space

$$\begin{aligned} b_k(\check{\mathbf{x}}^t, \check{\mathbf{x}}^p) &= p(\mathbf{x}_k = (\check{\mathbf{x}}^t, \check{\mathbf{x}}^p)) && (7) \\ &= b^t_k(\check{\mathbf{x}}^t) \cdot \delta_{\mathbf{x}^p_k}(\check{\mathbf{x}}^p) && (8) \end{aligned}$$

also contains the position of the platform. The platform position is fully observable, and therefore given by the Dirac delta centered on the true platform position. The localization process continues until the expected root-mean-squared error (RMSE) of the current estimate is below a threshold μ_T. We denote the index of termination by K, which means K is the lowest integer with $\mu(b_K) \leq \mu_T$. K is a random variable, dependent on the received measurements.

The control problem is then to find a policy

$$\mathbf{a}_k = \pi(b_k) \tag{9}$$

which at each decision step minimizes the expected time until localization for the current belief. This time corresponds to the expected future cost and is formalized by the value

$$V^\pi(b_k) = \mathbb{E}\left[\sum_{i \geq k}^{K-1} c(\mathbf{x}^p_i, \pi(b_i))\right] \tag{10}$$

of a belief b_k, when following policy π. The expectation goes over all future measurements $z_{k+1:K}$ and possible target positions $\mathbf{x}^t \sim b^t_k$. The next section describes a planning algorithm for this problem.

III. Path Planning Algorithm

A. Localizer

The bearing measurements of the target are fused to a target position estimate, using a grid-based discrete Bayes filter. The grid is initialized with a uniform prior and a side-length of 100 cells for the larger axis. Each measurement updates the density estimate in each cell, using a normal distributed likelihood. As the target estimate becomes more accurate, the grid is adaptively resized, to achieve a more fine-grained resolution. Details about the localizer can be found in [14]. The point estimate is extracted based on the cell with the maximum a posteriori density. If a measurement is an outlier, in the sense that the 4σ cones of all measurements do not intersect, the localization process is terminated with an error.

B. Policy rollout based planner

The path planning algorithm used in this paper is based on the policy rollout principle. Policy rollout [16] is an approximate dynamic programming technique, which estimates the future cost of an action by the expected value of the next belief, when following a base policy π^B in future decisions. Then the rollout policy π^R is given by

$$\begin{aligned} \pi^R(b_k) &= \underset{\mathbf{a} \in \mathcal{A}_k}{\operatorname{argmin}}\ \mathbb{E}\left[c(\mathbf{x}^p_k, \mathbf{a}) + V^{\pi^B}(b_{k+1})\right] && (11) \\ &= \underset{\mathbf{a} \in \mathcal{A}_k}{\operatorname{argmin}}\ Q^{\pi^B}(b_k, \mathbf{a}) \ . && (12) \end{aligned}$$

The action value Q^{π^B} represents the expected cost when performing an action and afterwards following the base policy in all subsequent steps until the objective is achieved. In our case, the cost of the immediate action is deterministic and the expected value of $V^{\pi^B}(b_{k+1})$ can be computed using samples, leading to the following Monte Carlo estimate of Q^{π^B}:

$$Q^{\pi^B}(b_k, \mathbf{a}) = c(\mathbf{x}^p_k, \mathbf{a}) + \frac{1}{N_a} \sum_{j=0}^{N_a - 1} \sum_{i > k}^{K_{aj}-1} c(\mathbf{x}^p_{aji}, \pi^B(b_{aji})) \ . \tag{13}$$

Here N_a is the number of Monte Carlo samples used to evaluate action $\mathbf{a}$. We use a discretized action set $\mathcal{A}_k$, and the sequential halving algorithm, to determine how often each action is sampled.

The sequential halving algorithm is an algorithm to find the best action of a finite set of alternatives, when only noisy samples of the action values are available [17]. It works by successively eliminating actions over multiple rounds and focuses the sampling on the promising ones. In the first round it samples each action. Then the worse half of the actions is removed. In the second round only the remaining actions are sampled, and again the worse half is removed. This continues until only a single action is remaining. The total budget N is distributed evenly on the rounds, which means that in later rounds more samples are used for each remaining action.

When predicting the future costs, the index of the state $\mathbf{x}_{aji}$ and belief b_{aji} refers to decision step i in the j-th rollout, when evaluating action $\mathbf{a}$. We use common random numbers (CRNs) to evaluate the action value, i.e. use the same target positions and measurement uncertainties to reduce the sampling variances. Details can be found in [14] and [15].

C. Restrictions to the action space

Due to the elevation dependency of the antenna pattern, a measurement taken close to the target is extremely unreliable. We consider this restriction in the planner by limiting $\mathcal{A}_k$ to actions sufficiently distant from the probable target positions. To determine probable target positions, a convex hull $\mathcal{C}^{95}(b_k)$ is computed, which contains 95% of the probability mass of the target estimate. This corresponds to the convex hull of the top 95% percentile of the grid cells, ordered by the probability mass they contain. Then the action set

$$\mathcal{A}_k = \left\{\mathbf{a} \in \mathcal{A} : \mathrm{d}(\mathbf{a}, \mathcal{C}^{95}(b_k)) > r_{\text{min}}\right\} \tag{14}$$

consists of those actions that keep at least a minimum distance from this convex hull. Here $\mathrm{d}(a, \mathcal{C})$ computes the minimal distance between action $\mathbf{a}$ and convex hull $\mathcal{C}$, $\mathcal{A}$ is the original action set, and r_{min} is a parameter. Figure 8 shows an example of the convex hull and a reduced action set. If $\mathcal{A}_k$ is empty or contains only one action, we reduce the threshold r_{min} until there are at least two actions. However, this situation did not occur in our experiments.

D. Base policy

The used base policy selects an action greedily from the action set (14). The action is chosen to maximize the determinant of the Fisher information, which is a common criterion in the literature [18], [19]. We approximate the grid-based belief b_k^t by a maximum a posteriori point estimate $\tilde{\mathbf{x}}_k^t$ and a covariance $\tilde{\mathbf{P}}_k^t$. Then the base policy

$$\pi^B(b_k) = \underset{\mathbf{a} \in \mathcal{A}_k}{\operatorname{argmax}} \det \left(\left(\tilde{\mathbf{P}}_k^t \right)^{-1} + \mathbf{J}(\mathbf{a}, \tilde{\mathbf{x}}_k^t) \right) \tag{15}$$

selects actions that maximize the determinant. The Fisher information $\mathbf{J}(\mathbf{a}, \tilde{\mathbf{x}}_k^t)$ of a target at position $\tilde{\mathbf{x}}_k^t = (\tilde{x}^t, \tilde{y}^t)$, measured from position $\mathbf{a} = (x^a, y^a)$ is given by

$$\mathbf{J}(\mathbf{a}, \tilde{\mathbf{x}}_k) = \frac{1}{\sigma^2 \tilde{r}^2} \begin{pmatrix} \sin\tilde{\theta} \cdot \sin\tilde{\theta} & -\frac{1}{2}\sin 2\tilde{\theta} \\ -\frac{1}{2}\sin 2\tilde{\theta} & \cos\tilde{\theta} \cdot \cos\tilde{\theta} \end{pmatrix} \tag{16}$$

where $\tilde{r} = \|\tilde{\mathbf{x}}_k^t - \mathbf{a}\|_2$ is the distance between the point estimate and the measurement position, and $\tilde{\theta} = \operatorname{atan2}(\tilde{y}^t - y^a, \tilde{x}^t - x^a)$ the angle. It should be noted that this base policy is only feasible because of the restricted action set, as otherwise the most informative action would always be the one that lies directly over the target estimate, with $\tilde{r} \to 0$.

E. Detection of measurement outliers

Even for measurements taken at a sufficient distance, outliers can occur. In most of the cases the measured power over angle has a pattern that strongly differs from the antenna pattern. We use the cosine similarity measure to filter out measurements that do not match the antenna pattern.

Therefore, the measured power over angle $\mathbf{m}_k \in \mathbb{R}^{180}$ is considered as a vector that contains the received power in 2° bins. With $\boldsymbol{\psi}(z_k) \in \mathbb{R}^{180}$ we denote the antenna pattern (see Figure 3), focused on the measured bearing z_k. Then the measurement quality is given by the scalar product

$$q(z_k) = \frac{\mathbf{m}}{\|\mathbf{m}\|_2} \cdot \frac{\boldsymbol{\psi}(z_k)}{\|\boldsymbol{\psi}(z_k)\|_2} \tag{17}$$

which is a value between -1 and 1, and quantifies how similar the measurement is to the antenna pattern. We accept a measurement if this value is above a threshold q_{min}, and otherwise perform a new measurement at the same position.

While the underlying direction finder [20] potentially returns multiple measurements, here we use the assumption that only a single target is present.

IV. Experimental Setup

A. System description

To evaluate the algorithm, we built an experimental system based on the AR200 UAV platform[1] (see Figure 1). The payload attached to the UAV consists of a Yagi antenna, a USRP B210 receiver, an Ellipse-D IMU, a Gigabyte Brix 8550 computer, and a Huawei e3372h LTE stick for communication. Figure 2 shows an image of the payload. The total weight of the payload amounts to 2.061 kg. Our emitter is based on a Raspberry Pi Zero and a LimeSDR software defined radio, which sends a continuous sine wave at 1984 MHz. This frequency was chosen due to an existing license to send on this frequency.

The AR200 UAV is controlled from a ground station. The path planner runs on a laptop and communicates with the ground station over a USB connection. In addition, the laptop is connected to the payload via an LTE connection.

The payload computer and laptop both run the robot operating system (ROS), version 16.04. The payload contains a node controlling the receiver, as well as a node performing the direction finding [20]. The extracted bearing measurements are sent to the laptop, where they are used to perform the path planning. The planner runs as a Java application on the laptop, and communicates with a Python-based ROS node over the ZeroMQ protocol.

In addition, we run a ROS node acting as a bridge to the UAV ground station. This node does not support a dedicated rotation of the UAV. We therefore perform the rotation by a sequence of three waypoints with different headings. This leads to a rotation based on three parts (see Figure 4). Before starting the rotation, the platform is kept stationary for one second to ensure it is in a stable position.

B. Experimental area

The experimental area is approximately 313 meters long and 120 meters wide. We discretized the area into possible waypoints, as can be seen in Figure 5. In total there are 112 actions from which the planner can choose, with a distance of 15 meters each. The ground station was placed at the origin.

[1] https://www.airrobot.de/

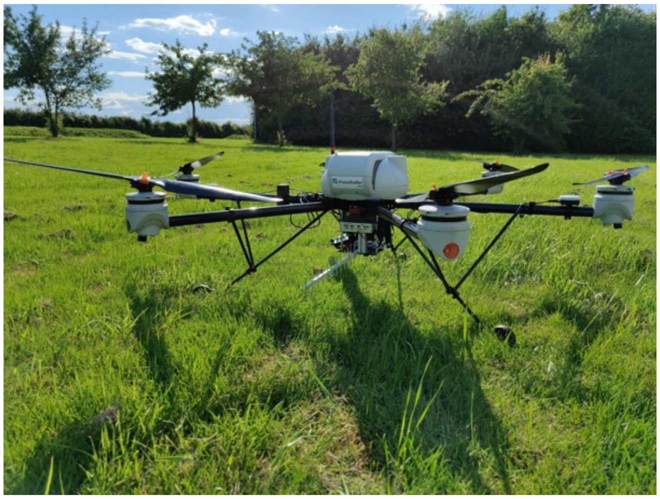

Fig. 1. Experimental platform with mounted payload.

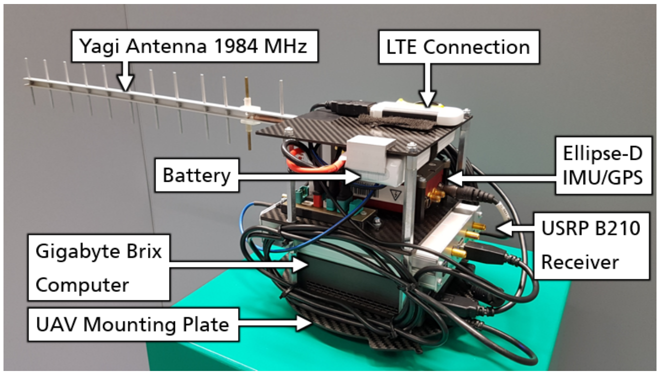

Fig. 2. Payload.

At the same location a human pilot was present, to perform landing and take-off, as well as to interrupt the UAV in case of emergencies. Note that the action set does not extent to the western border, as otherwise a tree line in the experimental area would have obstructed the line of sight between the human pilot and the UAV.

V. Experimental Results

A. *Performed flights*

We performed six flights on three different days to analyze the system. Table II shows an overview over those flights. Additional flights were taken previously for debugging and setup of the system. Flights 1-3 were made for system identification, consisting partially of manual measurements and partially of automatic localization attempts. Afterwards, the parameters of the planner were fixed to those shown in Table I. All flights were made at an altitude of approximately $20\,\mathrm{m}$. Flights 1-3 also included additional debugging. However, the direction finding algorithm was the same during flight 1-6, therefore those flights are included in the statistics about the direction finding performance.

For flights 4-6, we selected emitter positions B and C to increase the action space of the planner. As measurements should be taken at least $50\,\mathrm{m}$ away from the 95% confidence

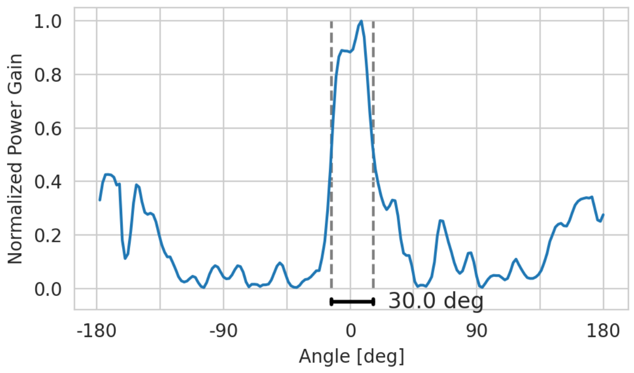

Fig. 3. Pattern of the antenna, mounted on the UAV.

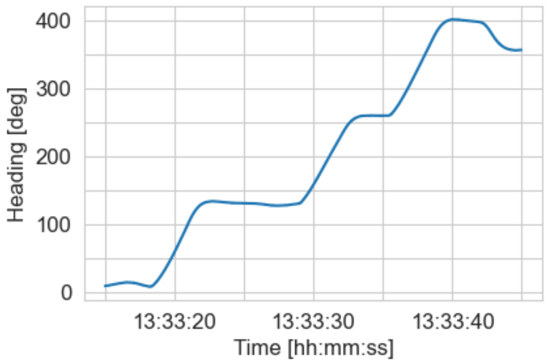

Fig. 4. Rotation during a measurement. The last rotation from 400 to 350 degrees is part of the movement to the next waypoint.

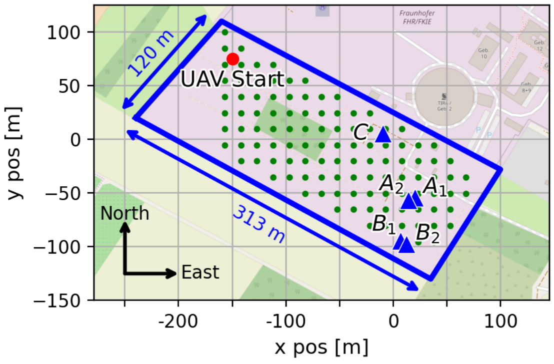

Fig. 5. Experimental area in an east, north, up (ENU) coordinate system. A, B, and C denote the two different emitter positions. The index denotes different positions from different trial days. The ground station was located at the coordinate origin. Green dots indicate the action set. The UAV start position corresponds to action $\mathbf{a}_0$. Map data © OpenStreetMap contributors.

TABLE I
PLANNER PARAMETERS

Parameter	Value	Description
σ	7°	Measurement standard deviation
μ_T	$10\,\mathrm{m}$	Localization accuracy threshold
$r_{\min}$	$50\,\mathrm{m}$	Minimal distance to uncertainty
v	$7\,\mathrm{m/s}$	Platform speed
t_M	$30\,\mathrm{s}$	Time of a measurement
N	1000	Total number of Monte Carlo samples
$q_{\min}$	0.7	Threshold on measurement quality

area, a centrally placed emitter limits the set of potential actions more than one on the border of the area. This was required because of the restricted size of our experimental area.

In total, eleven localization attempts were made, which can be seen in Table III. All of those had the UAV starting at the same position, shown in Figure 5. Seven localization attemps were fully successful, in the sense that the uncertainty was reduced successfully and the target was in the indicated uncertainty region. In two localization attempts (1 & 4) this was not the case, however, the target was at least somewhat close to the estimate. Two localization attempts (5 & 6) showed a completely wrong localization, which was due to the corresponding initial measurements were outliers. These outliers, together with the prior that the emitter is in the experimental area, reduced the uncertainty sufficiently that the emitter was considered localized after the first measurement. Therefore, the error criterion described in Section III-A was not triggered. Outliers are further discussed in the next section.

A localization typically required 4-6 measurements and took an average time of 3:53 minutes to localize emitter B and 3:02 (excluding 5 & 6) to localize emitter C. The paths taken by the UAV are shown in Figures 6 and 7 for emitter position B and C, respectively. It can be seen that the planner takes measurements from multiple directions, trying to be as close as possible to the target while keeping the $50\,\mathrm{m}$ distance. Figure 6 also illustrates how the planner adaptively reacts on different measurements. Measurements taken in flight 4 were all slightly more to the right, than measurements taken at flight 6. Therefore, in flight 4 the planner always flew to a waypoint at the north-eastern border of the experimental area, being convinced that this was sufficiently far from the target. In comparison, in flight 6 the planner always took a waypoint at the south-western border.

Figure 8 shows a single decision step of the planner. Only actions sufficiently far from the uncertainty estimate are evaluated. The actions are evaluated in multiple rounds, as the sequential halving algorithm focuses on the most promising actions. The color of the actions indicates the round in which they are eliminated. As the sequential halving algorithm eliminates half of the remaining actions in each round, this took six rounds for 56 viable actions in this step. It is visible that the planning algorithm quickly focuses on actions close to the target estimate. As the sequential halving algorithm uses the same number of samples in each round, the computation strongly focuses on the better actions. Actions eliminated in the first round were evaluated only with two samples, while the two last remaining actions were evaluated with in total 165 samples each.

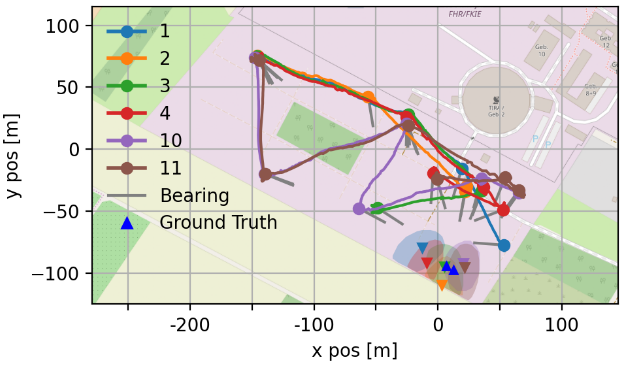

Fig. 6. Flight paths to localize emitter position B_1 (left, localization attempt 1-4) and B_2 (right, localization attempt 10-11). The emitter positions are indicated by the upright blue triangles. Localizations are indicated by the downward triangles. The shaded area around the estimates corresponds to the 95% probability mass of the uncertainty ($\mathcal{C}^{95}(b_K)$). Map data © OpenStreetMap contributors.

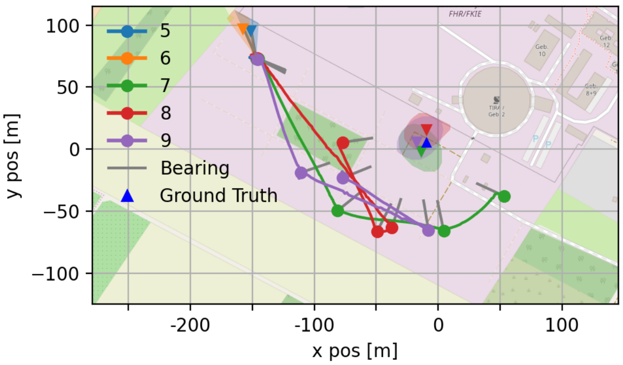

Fig. 7. Flight paths to localize emitter position C. The emitter position is indicated by an upright blue triangle. Localizations are indicated by the downward triangles. The shaded area around the estimates corresponds to the 95% probability mass of the uncertainty ($\mathcal{C}^{95}(b_K)$). Map data © OpenStreetMap contributors.

B. Bearing measurements

In total, the data from all flights contains 106 measurements. Of those, 19 are outliers, defined as having an error greater than 30°. Of the 13 measurements taken with a distance lower than $50\,\mathrm{m}$, 10 measurements are outliers indicating the high likelihood of a false measurement at short distances. Figure 9 shows the error of each measurement, dependent on the cosine similarity to the antenna pattern. Again, it can be seen that below a distance of $50\,\mathrm{m}$, measurements are mostly random, even if they have a high similarity measure. For measurements taken farther away, the threshold of 0.7 rejects most of the outliers. Together, this justifies the approach of a cautious planner, which does both, try to keep a sufficient distance from the target and

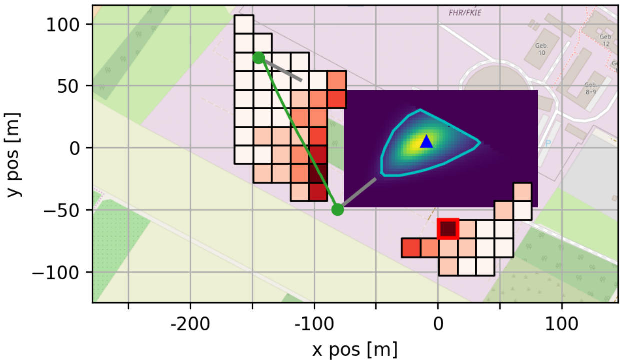

Fig. 8. Action evaluation in localization attempt 7, after the second measurement. The color of the action indicates in which of the six rounds it is eliminated. The selected action has a red border. The green line corresponds to the path of the UAV, gray are the measurements, the blue triangle is the ground truth and cyan is $\mathcal{C}^{95}(b_2)$. Map data © OpenStreetMap contributors.

TABLE II
FLIGHTS

Flight	Emitter	Duration	Avg. Wind	Max Wind	#Msr
1	A_1	11:56	-	-	12
2	A_1	20:14	-	-	13
3	A_2	21:13	5.6 km/h	14.3 km/h	32
4	B_1	19:05	6.5 km/h	15.0 km/h	19
5	C	18:17	2.6 km/h	5.6 km/h	16
6	B_2	14:23	3.2 km/h	-	14

Duration in [mm:ss]. No wind speed measurements were taken for flight 1 and 2. The wind measurement device showed an error message for the maximal wind speed on flight 6. Each group of two flights was made on a separate day. # Msr = Number of measurements. Flight 5 also contains two measurements not belonging to an automatic localization attempt, which therefore do not appear in Table III.

TABLE III
AUTOMATIC LOCALIZATIONS

Loc.	Emitter	Fl.	T.u.l.	Error	Exp.Err.	$\mathcal{C}^{95}$	#Msr
1	B_1	4	3:18	24.34 m	9.88 m	-	4 (+1)
2	B_1	4	2:34	15.75 m	9.10 m	✓	4
3	B_1	4	3:39	1.61 m	9.80 m	✓	4 (+1)
4	B_1	4	3:52	15.94 m	8.31 m	-	5
5	C	5	0:23	168.27 m	7.91 m	-	1
6	C	5	0:21	174.27 m	8.97 m	-	1
7	C	5	2:56	7.76 m	9.65 m	✓	4
8	C	5	2:52	10.47 m	8.94 m	✓	4
9	C	5	3:18	7.98 m	9.89 m	✓	4
10	B_2	6	4:44	9.46 m	9.16 m	✓	6
11	B_2	6	5:12	9.10 m	9.76 m	✓	6 (+2)

Loc. = Localization attempt, Fl. = Flight number as in Table II, T.u.l. = Time until localization [mm:ss] from the start of the first measurement to the announcement of the completed localization, Error = Distance between the point estimate and the ground truth, Exp.Err. = Expected Error $\mu(b_K)$, $\mathcal{C}^{95}$ = Whether the target was in $\mathcal{C}^{95}(b_K)$, #Msr = Number of valid measurements, passed to the planner. In parentheses are measurements classified invalid. Grouped localization attempts correspond to the same flight.

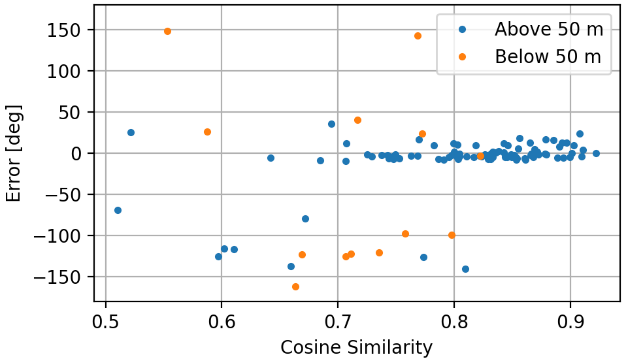

Fig. 9. Measurement error dependent on the cosine similarity outlier detection criterion.

reject measurements that do not match the antenna pattern. There are two outliers from a farther distance that are not detected. These correspond to the initial measurements in localization attempt 5 and 6. Both point directly into the corner of the experimental area, which was surrounded by a metal fence. A likely explanation is that the fence causes multipath reflections to occur.

Figure 10 shows the measurement errors of those measurements sufficiently far away ($> 50\,\mathrm{m}$), with a similarity measure > 0.7 and without those two outliers. On average the measurement error corresponds to the assumptions of the planner with a bias of $0.17°$ and a standard deviation of $7.47°$. However, the distribution seems to vary between different flights, with especially flight 4 having stronger errors. This is reflected in the results of localization attempts 1 and 4, where the target is not in the estimated uncertainty region $C^{95}_{(}b_K)$. It is unclear why the measurement quality differs from flight to flight. One possible explanation is a varying GPS quality, resulting in a worse GPS heading, and therefore direction estimate of the IMU. Our bearing estimator also explicitly tries to model the received pattern as a superposition of multiple signals and often returned multiple bearings. We then selected the bearing with the highest weight and discarded the others. A further tuning of the direction finder parameters [20], might force the direction finder to include less bearings of potentially higher quality.

The assumptions of the planner about the measurements were fixed after flights 1 and 2, and included an additional margin on the bearing error. The measurement threshold $q_{\min}$ was fixed after flight 3.

C. Time prediction accuracy

The planning algorithm predicts the time cost of different sensing actions to select those which lead to the minimal expected total cost. For this, it is important that the prediction model corresponds to the real cost of taking an action. In this section we analyze how accurate the planning algorithm models the time costs of the system. While the planner assumes linear movement, Figure 11 shows that a single movement of the UAV consists of different phases. First, the UAV rotates in the direction of its travel. Then it accelerates

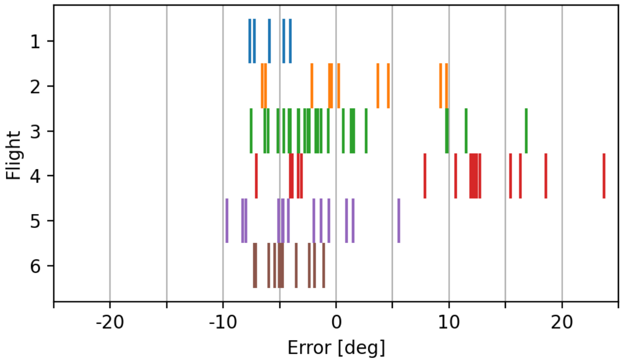

Fig. 10. Measurement error for measurements with $> 50\,\mathrm{m}$ distance and a cosine similarity measure > 0.7, excluding the two outliers. In total 81 measurements. A positive error indicates the measurement being right of the ground truth. The average errors amount to $\epsilon_1 = -5.90°$, $\epsilon_2 = 1.17°$, $\epsilon_3 = -0.24°$, $\epsilon_4 = 8.30°$, $\epsilon_5 = -3.13°$, $\epsilon_6 = -4.52°$ and the standard deviations to $\sigma_1 = 1.56°$, $\sigma_2 = 5.68°$, $\sigma_3 = 6.09°$, $\sigma_4 = 9.48°$, $\sigma_5 = 4.36°$, $\sigma_6 = 1.96°$.

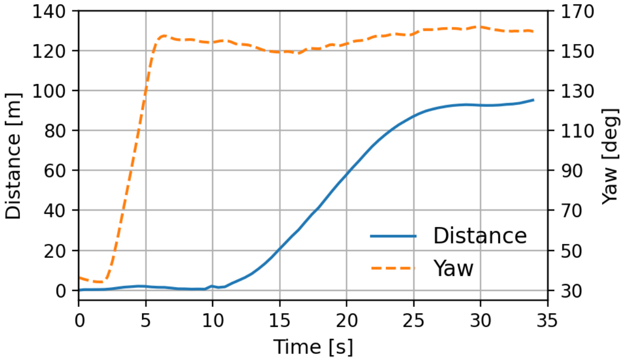

Fig. 11. Orientation and distance traveled during a movement phase.

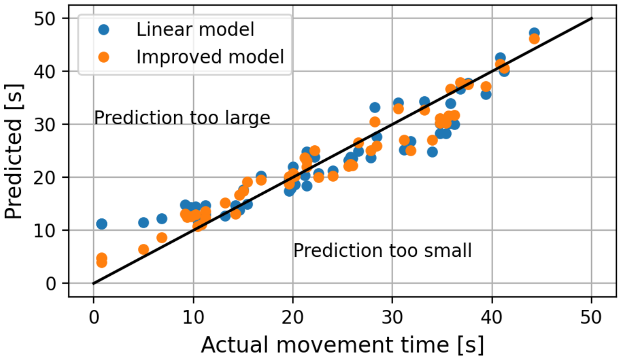

Fig. 12. Comparison between the prediction of movement time and the actual time. The linear model is a best-fit with $v = 7$ m/s. The improved model considers also acceleration and rotation.

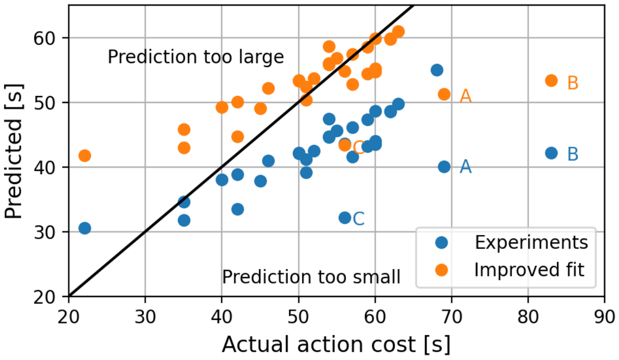

Fig. 13. Comparison between the predicted time for a sensing action, and the actual time. The improved fit assumes a longer measurement time $t'_M = 41.2\,\mathrm{s}$.

until its maximal speed, keeps this speed for some time, and decelerates before reaching the target position.

Figure 12 shows the accuracy of the predictions using a linear model with $7\,\mathrm{m/s}$ speed and a constant offset of $11.2\,\mathrm{s}$ (best fit). The RMSE of this linear model is $4.79\,\mathrm{s}$. We also developed an improved model after the experiments, which assumes that the UAV first rotates with 30°/s into its target orientation, then constantly accelerates with $1\,\mathrm{m/s^2}$ until either half of the way or $v_{\max} = 8.0\,\mathrm{m/s}$, decelerates before the target position and has an additional constant offset of $3\,\mathrm{s}$, covering additional stabilizing maneuvers. The RMSE of this model was $3.02\,\mathrm{s}$. While the more complex model is more accurate, the error of a linear model is not significantly higher. However, the linear approximation has a systematic error and overestimates consistently the travel costs of measurements with a short travel time, due to the constant offset.

Taking a measurement requires $21.4\,\mathrm{s}$ on average, with $2.9\,\mathrm{s}$ standard deviation. As we changed some termination criteria of the measurement rotation during flight 1 and 2, this average is computed using only flights 3 to 6. An execution of the planner requires $0.56\,\mathrm{s}$, with $0.35\,\mathrm{s}$ standard deviation.

Figure 13 shows the full end-to-end time prediction of a sensing action and the corresponding actual time. The RMSE of the prediction amounts to $14.1\,\mathrm{s}$. An improved fit uses $t'_M = 41.2\,\mathrm{s}$, which leads to an RMSE of $8.5\,\mathrm{s}$. One can see that the duration of short actions is overestimated, which mirrors the behavior seen in Figure 12 and indicates that a better movement model would improve the end-to-end time prediction, as well. However, in our implementation the planner had no access to the angle of the UAV. There are some outliers (A-C) visible in the plot, which correspond to planner steps with miss-detections and an additional measurement, making them approximately $21.4\,\mathrm{s}$ longer than expected. The fourth miss-detection (see Table III) happened at the begin of localization attempt 11, and is therefore not part of a planning prediction.

Overall, a linear prediction model seems to be an acceptable approximation, however, overestimates the costs of closer actions. An improved model would potentially make those actions more attractive to the planner.

VI. Conclusion and Future Work

In this paper we presented a system to localize an RF emitter, autonomously with a UAV. The control algorithm not only considers the next measurement positions, but performs a non-myopic planning to minimize the time until localization. It is based on a previously developed algorithm, which showed promising results in simulations. When adapting the algorithm to a real system, we encountered two major challenges: First, it is required that the algorithm keeps a minimal distance from the emitter. Second, occasionally bad measurements occur, which need to be filtered from the system.

We solved the first problem by computing a convex hull that contains 95% of the probability mass of the target position estimate. Then the planner is limited to those actions that are sufficiently distant from the convex hull. For the second problem, we reject all measurements which do not match the antenna pattern. This decision is made based on the cosine similarity measure.

We evaluated the algorithm and system in a total of six flights, and eleven autonomous localization attempts. From those eleven localization attempts, seven were fully successful, with the true emitter position in the indicated uncertainty region. Two flights led to a point estimate close to the emitter, but with the emitter outside of the uncertainty estimate. Two flights led to a wrong estimate, due to an outlier in the measurements. A probable explanation for this is a multipath reflection. We compared the models the planner uses for lookahead planning, and found out that they tend to overestimate the cost of close-by measurements. However, a more complex movement model can compensate for those differences. A large amount of unpredictability comes from the measurements, which are only approximately normal distributed and therefore hard to account for.

In future work it would be interesting to further analyze the measurement process. One question would be whether the variation of the measurement error distribution between flights can be reduced. It would also be of interest whether the outlier-detection can be improved or alternative ways to mitigate outliers can be found.

Also experiments on a less restricted area would be of interest. Finally, it would be interesting to experimentally compare the planner with a myopic one. This would show whether the advantages of non-myopic planning transfer from simulations to the real world.

Acknowledgements

We thank our colleagues, who helped with the execution of those experiments: Janning Nettekoven for assembling the payload, Manfred Okum for producing the Yagi antenna, Fahmi Rouatbi for help with the Ellipse-D and the UAV, Torsten Fiolka for help with the UAV and hardware recommendations, Matthias Mandt and Benjamin Knödler for help with RF measurements, Jannik Springer for support with an RF signal generator, and Ulrich Engel and Lars Brötje for help with the GPS.

References

[1] H. V. Nguyen, F. Chen, J. Chesser, H. Rezatofighi, and D. Ranasinghe, "LAVAPilot: Lightweight UAV trajectory planner with situational awareness for embedded autonomy to track and locate radio-tags," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2020, pp. 2488–2495.

[2] H. V. Nguyen, M. Chesser, L. P. Koh, S. H. Rezatofighi, and D. C. Ranasinghe, "TrackerBots: Autonomous unmanned aerial vehicle for real-time localization and tracking of multiple radio-tagged animals," *Journal of Field Robotics*, vol. 36, no. 3, pp. 617–635, 2019.

[3] M. Vrba, J. Pogran, V. Pritzl, V. Spurný, and M. Saska, "Real-time localization of transmission sources using a formation of micro aerial vehicles," in *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, Irkutsk, Russia, 2019, pp. 203–208.

[4] O. Cliff, R. Fitch, S. Sukkarieh, D. Saunders, and R. Heinsohn, "Online localization of radio-tagged wildlife with an autonomous aerial robot system," in *Robotics: Science and Systems XI*. Rome, Italy: Robotics: Science and Systems Foundation, 2015.

[5] J. Graefenstein, A. Albert, P. Biber, and A. Schilling, "Wireless node localization based on RSSI using a rotating antenna on a mobile robot," in *Proceedings of the 6th Workshop on Positioning, Navigation and Communication (WPNC)*, Hannover, Germany, 2009, pp. 253–259.

[6] J. T. Isaacs, F. Quitin, L. R. García Carrillo, U. Madhow, and J. P. Hespanha, "Quadrotor control for RF source localization and tracking," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, USA, 2014, pp. 244–252.

[7] J. Vander Hook, P. Tokekar, and V. Isler, "Cautious greedy strategy for bearing-only active localization: Analysis and field experiments," *Journal of Field Robotics*, vol. 31, no. 2, pp. 296–318, 2014.

[8] ——, "Algorithms for cooperative active localization of static targets with mobile bearing sensors under communication constraints," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 864–876, 2015.

[9] S. Venkateswaran, J. T. Isaacs, K. Fregene, R. Ratmansky, B. M. Sadler, J. P. Hespanha, and U. Madhow, "RF source-seeking by a micro aerial vehicle using rotation-based angle of arrival estimates," in *American Control Conference (ACC)*, Washington, DC, USA, 2013, pp. 2581–2587.

[10] K. Vonehr, S. Hilaski, B. E. Dunne, and J. Ward, "Software defined radio for direction-finding in UAV wildlife tracking," in *IEEE International Conference on Electro Information Technology*, Grand Forks, ND, USA, 2016, pp. 464–469.

[11] L. Dressel and M. J. Kochenderfer, "Pseudo-bearing measurements for improved localization of radio sources with multirotor UAVs," in *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018, pp. 6560–6565.

[12] F. Hoffmann, H. Schily, A. Charlish, M. Ritchie, and H. Griffiths, "A rollout based path planner for emitter localization," in *22nd International Conference on Information Fusion (FUSION)*, Ottawa, Canada, 2019.

[13] T. Gerlach, F. Hoffmann, and A. Charlish, "Policy rollout action selection with knowledge gradient for sensor path planning," in *24th International Conference on Information Fusion (FUSION)*, Sun City, South Africa, 2021.

[14] F. Hoffmann, A. Charlish, M. Ritchie, and H. Griffiths, "Policy rollout action selection in continuous domains for sensor path planning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2247–2264, 2021.

[15] F. Hoffmann, "Sensor path planning for emitter localization," Ph.D. dissertation, University College London, 2023.

[16] D. P. Bertsekas and D. A. Castañón, "Rollout algorithms for stochastic scheduling problems," *Journal of Heuristics*, vol. 5, no. 1, pp. 89–108, 1999.

[17] Z. Karnin, T. Koren, and O. Somekh, "Almost optimal exploration in multi-armed bandits," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, vol. 28, 2013, pp. 1238–1246.

[18] S. E. Hammel, P.-T. Liu, E. J. Hilliard, and K. F. Gong, "Optimal observer motion for localization with bearing measurements," *Computers & Mathematics with Applications*, vol. 18, no. 1-3, pp. 171–180, 1989.

[19] Y. Oshman and P. Davidson, "Optimization of observer trajectories for bearings-only target localization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, pp. 892–902, 1999.

[20] M. Krestel, F. Hoffmann, H. Schily, A. Charlish, and S. Rau, "Passive emitter direction finding using a single antenna and compressed sensing," in *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, 2019, pp. 1–5.