IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Path and trajectory planning of a tethered UAV-UGV marsupial robotic system

S. Martínez-Rozas[1], D. Alejo[2], F. Caballero[2] and L. Merino[3]

***Abstract*—This letter addresses the problem of trajectory planning in a marsupial robotic system consisting of an unmanned aerial vehicle (UAV) linked to an unmanned ground vehicle (UGV) through a non-taut tether with controllable length. To the best of our knowledge, this is the first method that addresses the trajectory planning of a marsupial UGV-UAV with a non-taut tether. The objective is to determine a synchronized collision-free trajectory for the three marsupial system agents: UAV, UGV, and tether. First, we present a path planning solution based on optimal Rapidly-exploring Random Trees (RRT*) with novel sampling and steering techniques to speed-up the computation. This algorithm is able to obtain collision-free paths for the UAV and the UGV, taking into account the 3D environment and the tether. Then, the letter presents a trajectory planner based on non-linear least squares. The optimizer takes into account aspects not considered in the path planning, like temporal constraints of the motion imposed by limits on the velocities and accelerations of the robots, or raising the tether's clearance. Simulated and field test results demonstrate that the approach generates obstacle-free, smooth, and feasible trajectories for the marsupial system.**

***Index Terms*—Motion and Path Planning; Aerial Systems: Applications**

I. Introduction

A marsupial multi-robot configuration [1] consists of a system in which one robot carries and can deploy one or several other robots. One example is the configuration in which an Unmanned Ground Vehicle (UGV) carries an Unmanned Aerial Vehicle (UAV), which can take off and possibly land back on the UGV when needed. This configuration can be used to combine the strengths of UGVs and UAVs. This UGV-UAV marsupial configuration was used by the Team CSIRO Data61 [2], reaching second place in the DARPA 2021 Subterranean Challenge [3]. Another notable example is the Mars 2020 rover and helicopter [4]. The helicopter augments the capabilities of the rover, exploring large areas faster than the rover, providing reconnaissance on target locations and safe to traverse routes.

Manuscript received: Mar., 28, 2022; Revised: Apr., 22, 2023; Accepted: July, 10, 2023.

This paper was recommended for publication by Editor Pauline Pounds upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the grants INSERTION PID2021-127648OB-C31 and RATEC PDC2022-133643-C21, funded by MCIN/AEI/10.13039/501100011033 and "European Union NextGenerationEU/PRTR".

[1]S. Martínez-Rozas, ORCID 0000-0002-3289-3371, is with Universidad de Antofagasta, Chile. Email: simon.martinez@uantof.cl

[2]D. Alejo (ORCID 0000-0002-3289-3371) and F. Caballero (ORCID 0000-0002-3289-3371) are with Service Robotics Laboratory, Universidad de Sevilla, Spain. Email: dalejo,fcaballero@us.es

[3]L. Merino is with Service Robotics Laboratory, Universidad Pablo de Olavide, Seville, Spain. lmercab@upo.es

Digital Object Identifier (DOI): see top of this page.

The flight time of small UAVs (few tens of minutes) is one of its main limiting factors. To increase their endurance, tethered UAVs fixed to a base station have been proposed [5], but the use of a tether limits the UAV range to a great extent. On the other hand, UGVs have longer autonomy, but they may not be able to reach certain areas of interest.

In this letter, we consider a tethered marsupial system, composed by a UAV attached to a UGV with a power cable, see Fig. 1. This configuration enables powering the UAV from the UGV, increasing its endurance and flexibility over systems tethered to a fixed position. The autonomous operation of the system requires considering the UGV-UAV-tether configuration as a whole for planning.

Thus, our goal is the development of motion planning modules for a mobile UGV-UAV-tether system. To the best of our knowledge, this is the first method that addresses the trajectory planning of a marsupial UGV-UAV with non-taut tether. The complexity of this planning problem lies in the dimension of the configuration space, that is, the positions of UAV and UGV, and the tether length. Furthermore, besides considering the collisions of each vehicle, we should also take into consideration the tether. This fact implies a large number of restrictions for the calculation of the trajectory. In our method, we do not assume a taut tether, considering the length of the tie as a control variable. This, together with the position of the UAV and UGV, allows us to be aware of the state of the tie through the mechanical model of the catenary. The main contributions of the letter are:

- A path planning method based on RRT* that generates collision-free paths for UAV and UGV, and lengths of the tether, to achieve a UAV goal configuration.
- A trajectory planning method based on non-linear optimization that considers smoothness, velocity and acceleration constraints, and optimizes the tether configuration to maximize clearance.

We present real experiments of an autonomous tethered UAV-UGV system that show how the provided trajectories are feasible. The letter is organized as follows. Section II analyzes the existing works in the literature addressing similar problems. Then, Section III formalizes the problem to solve in this letter. The proposed method for path planning is described in Section IV. This method is used as initial solution for the non-linear optimization problem presented in Section V for trajectory planning. The experimental results in simulated and real environments are discussed in Sections VI and VII, respectively. Finally, the conclusions and future research directions are detailed in Section VIII.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

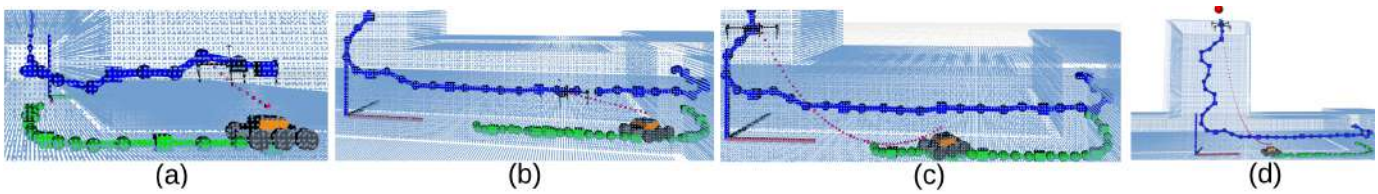

Fig. 1. Example of the developed method applied to a Marsupial robotic system. The trajectories of the UAV, UGV are represented in blue and in green, respectively. The tether in a configuration is represented in red. From top to bottom: a) starting configuration. b) configuration after the UAV and UGV turning inside the tunnel; c) the UGV is close to arrive to its final position. d) UAV and UGV in their final positions.

II. Related Work

Research on tethered systems can be found for Unmanned Underwater Vehicles (UUVs) [6], as the tether is used as a safety recovery device and as a communication link. However, it is usually assumed that the tethers move in free space and thus collisions due to tethers are not considered. The use of tethers has also been applied to UGVs for exploration. In [7] a tether is used to anchor the UGV to objects in the environment in order to explore a steep terrain. In [8] a UGV is tethered to a UAV, used as environment sensing assistance and also as an anchor to structures for climbing steep terrain. However, neither of those approaches consider obstacle avoidance for the tether.

In contrast, works on motion planning for tethered UAVs and/or tethered UGV-UAV configurations are scarce. In [5], a power-tethered UAV-UGV team is presented for tasks that require longer operation and increased inspection capabilities. The authors use two independent two-dimensional planners, incorporating the distinct navigation constraints and perception capabilities of each vehicle, the tether constraint, using the RRT* algorithm for path-planning. However, the planning system operates in 2-D planes (the UGV at ground level and the UAV at a given altitude), and it does not consider the length of the tether as a control variable.

Other approaches that consider aerial robots linked with a tether usually focus on control or estimation aspects. In [9], the authors present a system in which a human is physically connected to an aerial vehicle by means of a cable. A human-state aware controller allows the robot to pull the human toward the desired position including human velocity feedback. In [10], the use of a taut tether is proposed to increase the stability and safety in landing maneuvers. In [11], the authors consider also a tether to increase the stability when a UAV flights in confined spaces. In [12], the performance of several flight primitives for a tethered UAV is analyzed. The system controls the tether length, but assumes a taut tether to a fixed point. This is a limitation, as it forces the UAV to only access areas with a direct line of sight (LoS) from the fixed point. In [13], the hardware design of a tethered marsupial UGV-UAV platform is described, including the procedures for autonomous landing and a mission planner for pillar degradation analysis in underground mines. However, the tether is assumed to be taut and the planner generates trajectories with sweeping patterns only for the UAV system.

A tether state estimation technique is presented in [14], which describes a relative localization system for a marsupial tethered UGV-UAV pair. The estimation of the tether angles, as measured by the cable's provider SDK, and a catenary model [15] are used to estimate the relative position of the UAV with respect to the UGV. The same authors propose in [16] a PRM planner in which reachable space is constrained by the tether in two ways: avoiding any contacts of the tether with obstacles, or allowing up to two different contact points to extend the effective range. Nevertheless, again the planner assumes a tensed tether at any time, enabling collision checking based on ray-tracing. In [17] it is presented a tethered UAV-UGV team that navigates through unstructured or confined spaces. The UAV navigates autonomously and performs as visual assistant for UGV, while UGV is tele-operated. This approach also assumes a taut tether.

Regarding path planning, [18] presents a method for autonomous exploration of unknown cavities in three dimensions (3D) that focuses on minimizing the distance traveled and the length of the unwound tether. The method uses a hierarchical framework composed of a global level of exploration path planning that solves a Traveling Salesman Problem (TSP) and a local level of path planning whose parameters make it possible to adjust the trade-off between the tether unwinding and the path cost. This approach also assumes the tether to be always in a taut condition.

In [19], we proposed a trajectory planning method for an UAV tied to a fixed element. The tether length is also used as a decision variable, and a loose tether and its catenary model is used to achieve configurations outside of the LoS of the fixing point. The current work extends this one by proposing solutions to the planning problem for the UGV-UAV-tether system as a whole. This letter presents both, a path and a trajectory planner, to solve such problem. To the best of our knowledge, there are no approaches in the state of the art dealing with path or trajectory planning for a tethered UGV-UAV system considering a controllable loose tether.

III. Problem Statement

We define a state in the state space as the combination of the position of the UGV $\mathbf{p}_g = (x_g, y_g, z_g)^T$, the UAV $\mathbf{p}_a = (x_a, y_a, z_a)^T$ and the tether length l. At any instant, the tether length must be longer than the distance from the UGV to the UAV ($l \geq \|\mathbf{p}_a - \mathbf{p}_g\|$). We obtain the shape of the tether using the catenary model [15], whose parameters can be computed with the Bisection Numerical Method.

The motion planning problem consists in determining the trajectory for the UGV $\mathbf{p}_g(t)$, the UAV $\mathbf{p}_a(t)$ and the tether

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

length $l(t)$ so that the UAV reaches a given goal position, avoiding obstacles and meeting the constraints of the system.

We discretize the trajectories, so that the states of our problem become the set:

$$O = \{\mathbf{p}_g^i, \mathbf{p}_a^i, l^i, \Delta t^i\}_{i=1,\dots,n} \tag{1}$$

where i is a timestep of the trajectory and n is the total number of timesteps. $\Delta t^i = t^i - t^{i-1}$ is the time increment between steps i and $i-1$. This value is the same for UGV and UAV trajectories. For each $\mathbf{p}_a^i$, $\mathbf{p}_g^i$ and l^i, there is a tether configuration T^i, given by the catenary model mentioned above. When needed, we discretize it into a set of m positions $\mathbf{p}_t = (x_t, y_t, z_t)$:

$$T^i = \{\mathbf{p}_t^j\}_{j=1,\dots,m} \tag{2}$$

IV. Path Planning of a Tethered UAV-UGV System Using RRT*

In our first method, we disregard the temporal aspects and, thus, the dynamic constraints of the system, tackling the path planning problem. The objective is to determine the sequence of positions for the UAV and the UGV, and the adequate tether lengths at each step, to attain the UAV goal.

Our approach makes use of the RRT* algorithm [20] to solve the problem due to its ability to deal with high-dimensional spaces. However, we do not sample in seven dimensions (UGV, UAV, and tether length) to find the solution because of computation and problem domain reasons. Instead, the tether length is obtained procedurally in the RRT* `Steering` algorithm (see below), reducing the problem dimension to six. Our RRT* cost function is the weighted sum of the total length of the UAV and UGV paths, given that a new node is only accepted if there exists a collision-free catenary connecting the UGV with the UAV.

In order to properly consider the constraints derived from the tethered UAV and UGV configuration, we have adapted the procedures of RRT* algorithm [20], as described below.

A. RRT* problem setup

The RRT* will plan a path in a six-dimensional space composed by the UGV and UAV position $\mathbf{x} = \{\mathbf{p}_g, \mathbf{p}_a\}$, from an initial state $\mathbf{x}^i$ to a goal UAV state $\mathbf{x}^g$ in which only the aerial robot position is set (the final position of the UGV will depend on the UAV goal and the environment). The algorithm makes use of a 3D point cloud to represent the environment in which the robot system plans a path (see Fig. 2(a) as an example). This point cloud is processed for efficient obstacle representation for UAV and UGV. First, we use a method similar to [21] to perform a standard traversability analysis of the input 3D cloud to estimate the points that are traversable by the UGV, given the initial position. This traversable point cloud is used to sample UGV positions (see Fig. 2(b) as example). Second, we use a more convenient representation of the UAV environment (the full 3D point cloud) using a 3D grid containing the Euclidean Distance Field (EDF) of the environment [22]. The EDF is queried to get the distance to obstacles from any point, saving a great deal of computation effort. In our implementation, the EDF is computed offline, but we can use fast EDF implementations like FIESTA [23] for online computation.

B. Sampling process

When obtaining new nodes in free space through the `Sample` procedure, the UGV position $\mathbf{p}_g$ is sampled from the UGV's traversable points of the point cloud, and the UAV position $\mathbf{p}_a$ from the collision-free workspace. This sampling procedure allows us to plan UGV paths not only on flat surfaces but also with elevations (see Fig. 2(d)).

C. Getting the nearest node

To determine the nearest neighbor node in the current tree from a new node, the `Nearest` procedure considers a weighted sum of the Euclidean distances between the UGV and UAV positions of each node, and UGV orientation (UAV orientation is not considered because it is holonomic).

In terms of energy, the maneuvers of the marsupial robot system (once the UAV is flying) can be ordered from higher to lower consumption as follows: UAV translation, UAV hovering, UGV translation and stopped UGV [24]. However, since we aim the UAV to reach the goal, our approach prioritizes the movement of the UAV over the UGV (avoiding hovering). Thus, we apply a greater weight factor to the UGV distance. The nearest node is the one with the lowest cost.

D. checkCatenary algorithm

We include this algorithm in the `Steering` procedure. It checks if there exists a collision-free tether connecting the UGV to the UAV. The algorithm starts with the minimum tether length, i.e. straight line, and increases the length at fixed intervals until a collision-free tether is found or the maximum length is reached. If the collision-free tether is found, the states $\mathbf{p}_a^i$ and $\mathbf{p}_g^i$ are considered as feasible for the length l^i, and the algorithm returns `true`.

E. Steering process

The `Steering` procedure, described in Algorithm 1, has been adapted in order to facilitate the RRT* tree growth, as well as minimizing energy consumption of the marsupial system as described in Section IV-C. According to this, three modes have been defined, from highest to lowest priority:

1) The first mode steers the UAV position component only, and the UGV position is fixed. If the UAV is steered without collision and `checkCatenary` is `true`, then the *new node* is saved.
2) If the first mode cannot provide a viable node, the second mode is executed. The second mode steers UGV and UAV positions in such a way that if their poses are feasible and also `checkCatenary` is `true`, the *new node* is saved. Otherwise, the third mode is executed.
3) The last mode just steers the UGV and considers the UAV fixed. Then, if the UGV pose is feasible and `checkCatenary` is `true`, the *new node* is saved.

Finally, if a *new node* was not found in any one of three modes, a new *random node* is calculated using `Sampling`.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

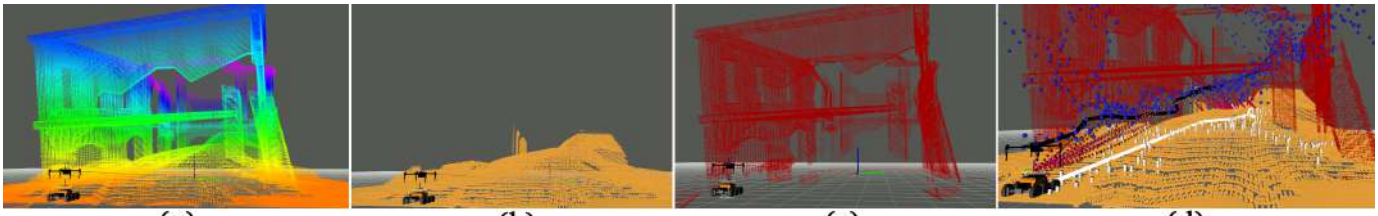

Fig. 2. (a) Example of full 3D point cloud (PC) for a scenario used for environment representation. (b) The traversable PC for the UGV computed from full cloud. (c) Obstacle PC for the UGV computed from full PC. (d) In white an example of sampling and computed path over traversable PC.

Algorithm 1: Steering process to get new node X_{new}. Function return TRUE if $\mathbf{X_{new}}$ is feasible. $\mathbf{X_{new}} = \{\mathbf{p}_g, \mathbf{p}_a\}$ save feasible position for UGV and UAV.

```
steering(X_nearest, X_rand)
    X_new ← steerUAV(X_nearest, X_rand)
    if obstaclesFree(X_nearest,X_new) &&
      checkCatenary(X_new) then
        return TRUE, X_new;
    end
    X_new ← steerUAVandUGV(X_nearest, X_rand)
    if obstaclesFree(X_nearest,X_new) &&
      checkCatenary(X_new) then
        return TRUE, X_new;
    end
    X_new ← steerUGV(X_nearest, X_rand)
    if obstacleFree(X_nearest,X_new) &&
      checkCatenary(X_new) then
        return TRUE, X_new;
    end
    return FALSE;
end
```

F. Obstacle Free test

The `ObstacleFree` process checks the feasibility (collision-free state) of the new node, and also if the system can be moved from the *nearest node* configuration to the *new node*. To this end, we interpolate the whole state between those two nodes and check for collisions and the existence of a catenary in each one of the interpolated states.

V. Trajectory Planning for a Tethered UAV-UGV System Using Non-Linear Optimization

The result of the path planning method from the previous section is a sequence of collision-free robot positions and tether lengths $\{\mathbf{p}_g^i, \mathbf{p}_a^i, l^i\}_{i=1,\dots,n}$, where n is the trajectory size. This sequence does not include the time-related information in (1), and neither considers time or safety constraints. In this section, we present a trajectory planning approach that improves that path by solving a non-linear optimization problem with dimension $8n$.

A. Initial trajectory estimation

The first step consists in adding time information to the initial path as the initial trajectory solution. For that, we take into account two main aspects.

The first one aims to adjust the initial sequence to have a set of equidistant waypoints. RRT* creates new nodes at a given distance (epsilon) from their parents, which is convenient for us, as we aim to sample the trajectory at regular intervals. However, because of our proposed `Steering` process (Section IV-E), we can get packed waypoints at the same position in the case of the UGV. To avoid this, whenever we detect some grouped points we spread them evenly to reach the next non-grouped point. Note that as we check for feasibility in the nodes connecting consecutive states (see Section IV-F), the new points are feasible. Furthermore, for each point, the tether length is calculated through the `checkCatenary` procedure.

Then, the Δt^i values are initialized. To do so, scalar constant speeds, v_g and v_a, are imposed over UGV and UAV path respectively. Both trajectories must use the same Δt^i to make UGV and UAV reach the waypoint at the same time. Thus, Δt^i value for each state of (1) is the largest value between $\|\mathbf{p}_g^i - \mathbf{p}_g^{i+1}\|/v_g$ and $\|\mathbf{p}_a^i - \mathbf{p}_a^{i+1}\|/v_a$.

B. Casting the problem as a sparse non-linear optimization

Based on the discretization of the states formulated in (1), our problem consists in determining the values of the variables in O that optimize the following function $f(O)$:

$$O^* = \arg\min_O f(O) = \arg\min_O \sum_{i,k} \gamma_k ||\delta_k^i(O)||^2 \qquad (3)$$

O^* denotes the optimized collision-free trajectory for UAV, UGV, and tether from start to goal configurations. γ_k is the weight factor for each component $\delta_k^i(O)$ (known as residual for each k constraint) of the objective function. Each component takes values between [0-1] and encodes a different constraint or optimization objective of our problem, and will be presented next. We include the constraints into cost functions as soft constrains, as including them as non-linear hard constraints would increase the complexity of the minimum search. The weights values are adjusted empirically and their selected values are shown in Section VI. Besides, each component should be evaluated in all the timesteps i of the trajectory . These components are local with respect to i, as they depend on a few consecutive states in general. Consequently, our optimization problem can be solved with non-linear sparse optimization algorithms. In particular, we use *Ceres-Solver* [25] as back-end.

C. Constraints and Objective Function

We represent the constraints in the problem as penalty costs in the objective function. The proposed constraints are related to the tether length, the limits on velocity and acceleration, and the equidistance among consecutive robot poses. In addition,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

we aim to maximize the distance from obstacles to each agent, minimize the execution time of the trajectory, and maximize its smoothness. We use the Cauchyloss (robust loss function) as a robust kernel in order to reduce the influence of outliers, which might appear in the map generation for instance, in the solution [26].

1) Equidistance among consecutive states: The optimization process might unevenly move UGV and/or UAV positions within the planning space in the search of the minimum cost. To avoid it, we force them to keep a given distance among states, ρ_{eg} for the UGV and ρ_{ea} for the UAV. They are obtained by dividing the distance between the initial and goal points along the path by the number of steps. In this section we formulate the UGV case, δ^i_{ea} is calculated similarly.

$$\delta^i_{eg} = \|\boldsymbol{p}^{i+1}_g - \boldsymbol{p}^i_g\| - \rho_{eg} \tag{4}$$

2) UAV/UGV Obstacle avoidance: This constraint penalizes the UAV/UGV states whose distance to the nearest obstacle $d^i_{oa}(d^i_{og})$ is closer than a safety distance $\rho_{oa}(\rho_{og})$, which depends on the size of the UAV/UGV.

$$\delta^i_{og} = \begin{cases} \rho_{og} - d^i_{og} & \text{, if } d^i_{og} < \rho_{og} \\ 0 & \text{, otherwise} \end{cases} \tag{5}$$

3) Tether obstacle avoidance: This constraint penalizes the proximity to obstacles of the m samples in which the tether is discretized. The tether state is computed by solving the catenary model for l^i between $\mathbf{p}^i_g$ and $\mathbf{p}^i_a$, according to (2). We compute the distance to the nearest obstacles of each sample of the tether, $d^i_{ot,j}$, and the residual as the sum of the inverse nearest distances. We increase the weight of those samples closer than a safety distance ρ_{ot} to guarantee higher costs in these cases using $\rho_j = \beta$, with $\beta >> 1$.

$$\delta^i_{ot} = \sum_{j=1}^{m} \frac{\rho_j}{d^i_{ot,j}},\ \rho_j = \begin{cases} 1 & \text{, if } d^i_{ot,j} > \rho_{ot} \\ \beta & \text{, otherwise} \end{cases} \tag{6}$$

4) UGV traversability: It is necessary for the UGV to remain in traversable areas. To this end, we impose a δ_{trav} penalty to the UGV states that are away a distance $d^i_{trav} > \rho_{trav}$ from the traversable area. Thus, d^i_{trav} is the distance to the closest point into the traversable point cloud.

$$\delta^i_{trav} = \begin{cases} d^i_{trav} - \rho_{trav} & \text{, if } d^i_{trav} > \rho_{trav} \\ 0 & \text{, otherwise} \end{cases} \tag{7}$$

5) Smoothness: This constraint is in charge of avoiding abrupt direction changes for UGV and UAV trajectories. From three consecutive positions, two vectors are calculated. The first between i and $i-1$ states, and the second between $i+1$ and i states. Then, we calculate the angle between these vectors, θ_g for the UGV, and θ_a for the UAV. The residuals δ_{sg} and δ_{sa} penalize the states whenever the angle between vectors exceeds a given threshold ρ_{sg} and ρ_{sa} respectively.

$$\cos\theta_g = \frac{(\mathbf{p}^i_g - \mathbf{p}^{i-1}_g)\cdot(\mathbf{p}^{i+1}_g - \mathbf{p}^i_g)}{\|\mathbf{p}^i_g - \mathbf{p}^{i-1}_g\|\|\mathbf{p}^{i+1}_g - \mathbf{p}^i_g\|} \tag{8}$$

$$\delta^i_{sg} = \begin{cases} 1 - \cos(\theta_g), & \text{if } |\theta_g| > \rho_{sg} \\ 1 - \cos(\rho_{sg}), & \text{otherwise} \end{cases} \tag{9}$$

6) Velocity: This constraint relates two consecutive positions and the timestep between them, for the UGV and the UAV. It keeps the speed for both the UGV and the UAV during the optimized trajectory as constant as possible. Then, the error δ^i_{vg} corresponds to the difference between the computed states velocity and the desired velocity ρ_{v_g}.

$$\delta^i_{vg} = v^i_g - \rho_{vg} \tag{10}$$

where $v^i_g = \frac{|\mathbf{p}^{i+1}_g - \mathbf{p}^i_g|}{\Delta t^{i+1}}$ is the average velocity between two consecutive poses for the UGV.

7) Acceleration: It relates three consecutive positions and their related time steps for the UGV and the UAV, so that the linear acceleration close to zero, minimizing control efforts.

$$\delta^i_{ag} = \frac{v^i_g - v^{i-1}_g}{\Delta t^i + \Delta t^{i+1}} \tag{11}$$

8) Unfeasible tether length: This constraint penalizes unfeasible tether lengths, that is, tethers with length l^i shorter than the Euclidean distance between $\mathbf{p}^i_g$ and $\mathbf{p}^i_a$, d^i_u.

$$\delta^i_u = \begin{cases} e^{d^i_u - l^i} - 1 & \text{, if } d^i_u > l^i \\ 0 & \text{, otherwise} \end{cases} \tag{12}$$

VI. Experimental Results in Simulation

Both methods, path and trajectory planners, have been implemented in C++, and the full approach as a planner in Robot Operating System (ROS). The source code is publicly available[1]. The *Ceres-Solver* [25] has been selected as the back-end to solve the optimization problem. We used a 9th gen Intel Core i7 running at 2.20GHz with 32 GB of RAM.

To validate the feasibility of the proposed approach, we have conceived a set of experiments in five synthetic scenarios (S1 to S5), with different degrees of complexity (see Fig. 3) and also field experiments . For better understanding, we have included an extended video [2] that includes results on each of the scenarios. Our methods can find solutions in environments where approaches that assume a taut tether and/or a fixed UGV pose would fail. As an example, let us consider the output of our method in S3, where the UAV must turn inside a corridor and then enter a chimney to reach the goal, see Fig. 1. For that, the UGV should enter the corridor, moving close to the chimney to let the UAV enter.

For each scenario, we compute a path using the RRT* method in batches of 500 iterations until a solution is found. Then, that solution is used as the initial guess in the optimization step. We set the maximum number of iterations to 50000, obtained by performing experiments in different scenarios, as RRT* usually succeeds before the 1000th iteration.

We test the methods in two different initial positions for each scenario. The maximum number of optimizer iterations is 1000. All the experiments have been executed 100 times with the same set of parameters detailed below. The weight factors, selected empirically, are in the $[0, 1]$ range.

[1]https://github.com/robotics-upo/marsupial_optimizer
[2]https://youtu.be/N-K3yT8Tsxw

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

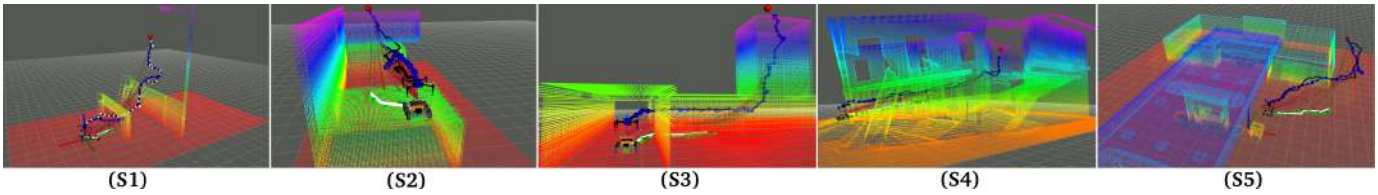

Fig. 3. Scenarios considered for validation.S1: Open/constrained space with arc as obstacle. S2: Narrow/constrained space with denied access to UGV. S3: Confined space with outlet duct for UAV. S4: Collapsed Fire Station. S5: Open space gas station.

- Weighting factors for UGV: γ_{eg} = 0.2, γ_{og} = 0.08, γ_{trav} = 0.5, γ_{sg} = 0.12, γ_{vg} = 0.05, γ_{ag} = 0.005.
- Weighting factors for UAV: γ_{ea} = 0.25, γ_{oa} = 0.08, γ_{sa} = 0.14, γ_{va} = 0.05, γ_{aa} = 0.005.
- Weighting factors for Tether: γ_{ot} = 0.25, γ_{u} = 0.1
- Equidistance threshold: ρ_{eg} and ρ_{ea} are computed based on the initial path.
- Collision threshold: ρ_{oa} = 1.2, ρ_{ot} = 0.1, ρ_{og} = 1.2.
- Traversability threshold: ρ_{trav} = 0.001.
- Smoothness threshold: $\rho_{sg} = \frac{\pi}{9}$, $\rho_{sa} = \frac{\pi}{9}$.
- Desired velocity (m/s): ρ_{vg} = 1.0, ρ_{va} = 1.0.
- Tether avoidance: β = 10.0 .

The results of the experiments are summarized and detailed in Tables I and II. In order to benchmark the solution, we use as baseline the value of the different metrics prior to optimization, that is, the solution of the RRT* detailed in Section IV. Next paragraphs evaluate the different metrics.

A. Feasibility

The result of the planners is considered as feasible when the computed path/trajectory is collision free for every agent of the system. This means that $d^i_{og} > \rho_{og}$ and $d^i_{oa} > \rho_{oa}$ for every UGV and UAV state respectively. In case of tether, $d^i_{ot,j} > \rho_{ot}$ for every sample in each tether configuration.

An important result is that the RRT* planner is able to find a solution in 100% of the tested cases. On the other hand, the feasibility of the trajectory planner is 86.6%. Please note that in 99% of the unfeasible solutions are due to tether collisions, mainly occurring in cluttered environments with obstacles above and below the tether. The constraints considered by the optimizer might move the solution far from the initial guess provided by RRT*. In addition, the tether obstacle avoidance constraint (6) can only be computed numerically, making its gradient more sensible to numerical issues. As a result, the optimizer can produce unfeasible solutions.

B. Trajectory length, elapsed time and smoothness

The trajectory planner does not minimize the trajectory length explicitly. The constraints influence the length of the optimized trajectories, possibly increasing their value w.r.t. the initial one. This is either by moving away from obstacles or by increasing the curvature of the trajectory to smooth it.

C. Distance to obstacles

The distance to obstacle statistics shown in Table II (DOI, DOO, DCOI, DCOO) for UGV and UAV confirm how the optimized trajectory tends to move away from obstacles when the trajectory itself is closer than the safety distance. For the tether, the DCOO min is on average above the considered safety value ρ_{ot}. The DCOI and DCOO mean values are similar for both methods.

D. Velocities and Accelerations

The results in Table I show that the optimized velocities and accelerations (VTO, ATO) are close to the desired values, which are 1.0 m/s and 0 m/s^2, respectively. There is a noticeable improvement on these values w.r.t. the initial ones.

E. Computation Time

Table I presents the computation time (TCI, TCO) required to compute the initial and optimized trajectories. The results indicate that the RRT* algorithm obtains a solution in around half a second in confined scenarios, while lasting up to 27.9 seconds in open spaces. On the other hand, the optimization algorithm requires longer times due to the need of repeatedly solving the transcendental catenary equations [15]. However, it is worth noting that CERES solver was configured with a single CPU thread. In this implementation, the computational time can be estimated as the single thread time divided by the number of threads used. That is, we can reduce it an order of magnitude by using ten CPU threads.

VII. Field Experiments

Once the method has been analyzed in different simulation scenarios, we proceed to validate the solution in a real marsupial system using state-of-the-art localization, navigation, and control approaches to execute the proposed trajectories.

The marsupial team is commanded to inspect the old theater building of the Pablo de Olavide University, Seville (Spain) (see Fig. 4.(a)). To this end, the UAV should go to an inspection point (IP) above the stage of the theatre, out of the reach of the UGV and not in its LoS. Therefore, the UGV should go to a position that allows the UAV to reach the IP. The experiment is shown in the attached video.

A. Marsupial System

The marsupial system consists of a M210 drone platform from DJI, an ARCO omnidirectional ground robot from ID-Mind Robotics, and a custom-made automated reel that adjusts the length of the tether. Each subsystem performs its tasks fully autonomously. The robots carry an OS1 LIDAR for localization and navigation, no external motion capture system was used. In contrast, we used our Direct LIDAR Localization method [27] to estimate their pose.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE I

RESULTS FOR FEASIBILITY, COMPUTE TIME AND TRAJECTORY TIME PARAMETERS IN SIMULATED ENVIRONMENTS. SI: SCENARIO; F: FEASIBILITY; TCI: TIME FOR INITIAL SOLUTION $[s]$; TCO: TIME FOR OPTIMIZED SOLUTION $[s]$; VTI: VELOCITY, INITIAL TRAJECTORY $[m/s]$; VTO: VELOCITY, OPTIMIZED TRAJECTORY $[m/s]$; ATI: ACCELERATION, INITIAL TRAJECTORY $[m/s^2]$; ATO: ACCELERATION, OPTIMIZED TRAJECTORY $[m/s^2]$

ALL				UGV								UAV							
SI	F	TCI	TCO	Mean VTI	Max VTI	Mean VTO	Max VTO	Mean ATI	Max ATI	Mean ATO	Max ATO	Mean VTI	Max VTI	Mean VTO	Max VTO	Mean ATI	Max ATI	Mean ATO	Max ATO
S1.1	86.0	5.1	465.9	0.37	0.78	0.36	1.07	-0.23	0.15	0.02	0.81	0.88	1	0.97	1.19	-0.74	0.25	-0.002	-0.09
S1.2	80.0	15.0	529.7	0.50	0.97	0.43	1.25	-0.26	0.22	0.02	0.78	0.87	1	0.91	1.19	-0.72	0.32	-0.002	-0.08
S2.1	94.5	0.4	123.6	0.11	0.31	0.13	0.91	-0.11	0.16	0.02	0.96	0.92	1	1.01	1.15	-0.71	0.06	0.002	0.01
S2.2	95.0	0.5	109.3	0.27	0.70	0.19	1.00	-0.15	0.15	0.03	1.09	0.92	1	0.99	1.09	-0.70	0.11	0.000	0.07
S3.1	84.8	0.5	213.6	0.43	0.90	0.50	1.07	-0.29	0.30	0.00	0.21	0.88	1	1.02	1.21	-0.73	0.20	0.003	0.16
S3.2	84.0	1.5	214.9	0.49	0.96	0.55	1.08	-0.31	0.34	0.00	-0.20	0.85	1	0.98	1.16	-0.69	0.26	0.004	0.18
S4.1	82.5	1.2	539.0	0.57	0.98	0.45	1.14	-0.33	0.24	0.02	1.00	0.88	1	0.94	1.12	-0.71	0.30	-0.002	-0.06
S4.2	76.0	12.8	708.6	0.43	0.96	0.47	1.32	-0.25	0.23	0.01	0.86	0.87	1	0.92	1.22	-0.70	0.40	-0.002	-0.32
S5.1	98.0	27.9	277.8	0.45	0.95	0.41	1.15	-0.23	0.25	0.03	1.07	0.87	1	0.97	1.27	-0.64	0.25	-0.004	-0.14
S5.2	95.0	26.0	447.7	0.67	1.00	0.57	1.08	-0.41	0.43	0.02	1.02	0.88	1	0.92	1.11	-0.68	0.39	-0.004	-0.12

TABLE II

RESULTS FOR TRAJECTORY LENGTH AND DISTANCE TO OBSTACLES IN SIMULATED ENVIRONMENTS. LIP: LENGTH, INITIAL PATH $[m]$; LTO: LENGTH, OPTIMIZED TRAJECTORY $[m]$; DOI: DISTANCE TO OBSTACLES, INITIAL PATH $[m]$; DOO: DISTANCE TO OBSTACLES, OPTIMIZED TRAJECTORY $[m]$; DCOI: DIST. CATENARY-OBSTACLE INITIAL SOLUTION $[m]$; DCOO: DIST. CATENARY-OBSTACLES, OPTIMIZED SOLUTION $[m]$

ALL	UGV						UAV						Tether			
SI	LIP	LTO	Mean DOI	Min DOI	Mean DOO	Min DOO	LIP	LTO	Mean DOI	Min DOI	Mean DOO	Min DOO	Mean DCOI	Min DCOI	Mean DCOO	Min DCOO
S1.1	6.3	6.3	2.26	1.45	2.28	1.55	14.6	14.2	1.83	0.65	1.87	0.80	1.15	0.31	1.14	0.25
S1.2	10.7	10.8	3.74	1.50	3.76	1.60	18.8	18.2	2.20	0.85	2.21	0.89	1.30	0.22	1.31	0.16
S2.1	1.5	1.6	0.65	0.56	0.69	0.56	13.1	12.7	1.10	0.59	1.19	0.76	0.71	0.13	0.72	0.10
S2.2	2.7	2.7	0.75	0.58	0.75	0.58	13.4	13.1	1.11	0.56	1.16	0.67	0.75	0.17	0.75	0.13
S3.1	8.6	8.9	0.88	0.50	0.98	0.62	17.7	17.6	0.69	0.51	0.81	0.56	0.68	0.20	0.67	0.13
S3.2	11.1	11.4	0.88	0.50	0.97	0.61	19.0	18.9	0.69	0.50	0.79	0.54	0.68	0.23	0.67	0.14
S4.1	10.8	10.8	2.12	1.39	2.12	1.40	20.9	20.7	1.09	0.53	1.16	0.65	0.80	0.22	0.74	0.14
S4.2	15.4	16.0	1.70	1.02	1.70	1.07	27.1	26.7	1.06	0.51	1.15	0.65	0.82	0.22	0.76	0.10
S5.1	9.4	9.2	2.24	0.70	2.26	0.76	18.5	18.1	1.71	0.72	1.74	0.83	1.12	0.30	1.09	0.23
S5.2	23.7	23.3	2.52	0.59	2.53	0.63	35.2	34.9	1.65	0.58	1.66	0.65	1.17	0.25	1.14	0.22

In order to track the trajectory computed by our approach, the marsupial system must perform a coordinated motion of the UGV, UAV, and automated reel. We opt for a loosely-coupled solution based on synchronization. Thus, new waypoints cannot be commanded until all the subsystems confirm reaching the current one. When a subsystem reaches its goal, it waits for the others in case of need.

For safety reasons, the maximum commanded speed has been limited to the slowest subsystem, which in this implementation is the automated reel. Thus, the planner has been parameterized as in Section VI, but setting ρ_{vg} and ρ_{va} to $0.25m/s$ to accommodate the solution to the reel dynamics.

Both robots implement a control scheme based on PIDs and trapezoidal velocity profile for waypoint tracking. The maximum velocity for the trapezoid is set to the planned ρ_{vg} and ρ_{va} respectively, but there are velocity ramps that slightly reduce the average speed during the maneuvers.

B. Summary of the results

A trajectory for the complete system was computed by the proposed method to guide the system from starting to goal points (see Fig. 4). This trajectory was successfully tracked by the marsupial system in five different experiments at a commanded speed of 0.25 m/s for both platforms, the desired speed (ρ_v) in the optimization process. Table III shows the maximum and average tracking errors in position, velocity and time of each subsystem for one of those experiments. The mean position tracking error of both platforms was below $0.06m$ in the experiments. The robots' control system implemented a reaching-waypoint-criteria of $0.1m$ around the objective. Thus, the waypoints located closer than $0.1m$ were discarded when calculating the errors.

The system successfully tracked the trajectory with mean errors below $0.08m/s$ in velocity and $0.08m/s^2$ in acceleration. The tether tracking errors were also slim during the whole experiment, about $0.01m$ in average. The maximum errors shown in Table III are mostly produced by the trapezoidal velocity profile scheme for trajectory tracking. The velocity ramps reduce the averaged robot velocity, producing deviations in velocity, acceleration, and time. In addition, the UGV performs a rotation in place when the angle to reach the waypoint exceeds a given threshold, increasing the execution time. This happened twice in the experiment.

All in all, the marsupial system was able to follow the plan without collisions with the tether, as shown in the video.

VIII. CONCLUSIONS AND FUTURE WORK

This paper presented a general path and trajectory planning algorithm for a marsupial system. Contrary to most existing approaches, our methods do not assume a taut tether, using the catenary model instead. In this way, we can consider both taut and loose configurations. The planning methods have been tested in simulation environments that need of UGV translation and a loose tether to reach the goal position.

Despite the complexity of the problem, our adapted RRT* algorithm provides solutions in few seconds, with a success rate of 100%. The optimization process generates safer

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

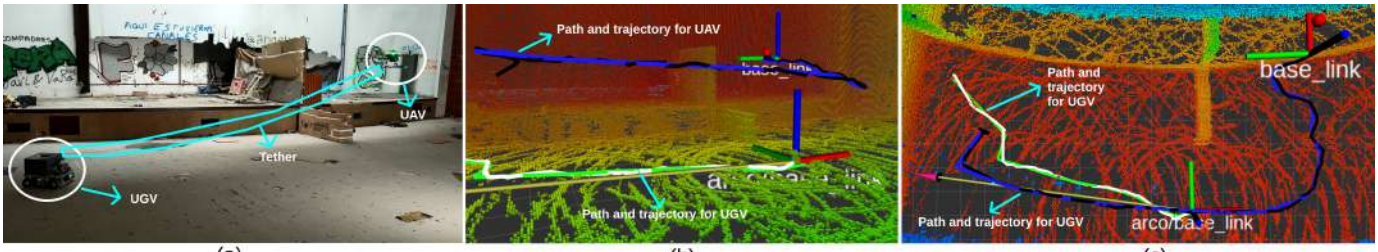

Fig. 4. Validation of the Experiments. (a) Our marsupial system during the experiments at the abandoned theatre. (b) Computed and executed trajectories of the UAV (in blue and black lines, respectively) and UGV (green and white) in the camera view. (c) Same representation in top view.

TABLE III
STATISTICAL ANALYSIS FOR TRAJECTORY TRACKING ERRORS (TIME $[s]$, VELOCITY $[m/s]$, ACCELERATION $[m/s^2]$)

	UGV				UAV					Tether
Measure	Position XY	Time	Vel.	Acc.	Position XY	Position Z	Time	Vel	Acc.	Length
Mean error	0.03	0.46	0.08	0.08	0.05	0.06	0.35	0.05	0.05	0.01
Max error	0.17	3.32	0.18	0.53	0.30	0.33	2.00	0.15	0.21	0.31

trajectories than planner in terms of distance to obstacles, especially related to UGV and UAV agents. Finally, we have executed a trajectory generated by the proposed method in field experiments, demonstrating its safety and feasibility.

Future work will consider the development of analytical approximations to speed up the computation of the state of the catenary model. Analytical approximations of the catenary will improve error gradient estimation. Finally, some tether collisions planning could be allowed, e.g. with the floor.

REFERENCES

[1] R. Murphy *et al.*, "Marsupial-like mobile robot societies," in *AGENTS '99*, 1999.
[2] N. Hudson *et al.*, "Heterogeneous ground and air platforms, homogeneous sensing: Team CSIRO data61's approach to the DARPA subterranean challenge," *Field Robotics*, vol. 2, no. 1, pp. 595–636, mar 2022. [Online]. Available: https://doi.org/10.55417%2Ffr.2022021
[3] DARPA, "DARPA SUBTERRANEAN CHALLENGE 2021," https://https://www.subtchallenge.com/index.html, 2021.
[4] H. F. Grip *et al.*, "Guidance and control for a mars helicopter," in *NASA*, 2018.
[5] C. Papachristos and A. Tzes, "The power-tethered uav-ugv team: A collaborative strategy for navigation in partially-mapped environments," in *22nd Med. Conf. Control and Aut.*, 2014, pp. 1153–1158.
[6] M. Laranjeira, C. Dune, and V. Hugel, "Catenary-based visual servoing for tether shape control between underwater vehicles," *Ocean Engineering*, vol. 200, p. 107018, 2020.
[7] P. McGarey, F. Pomerleau, and T. D. Barfoot, *System Design of a Tethered Robotic Explorer (TReX) for 3D Mapping of Steep Terrain and Harsh Environments*. Springer, 2016, pp. 267–281.
[8] T. Miki, P. Khrapchenkov, and K. Hori, "Uav/ugv autonomous cooperation: Uav assists ugv to climb a cliff by attaching a tether," in *2019 Int. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 8041–8047.
[9] M. Allenspach *et al.*, "Human-state-aware controller for a tethered aerial robot guiding a human by physical interaction," *IEEE Robotics and Automation Letters*, pp. 2827–2834, 2022.
[10] L. A. Sandino, D. Santamaria, M. Bejar, A. Viguria, K. Kondak, and A. Ollero, "Tether-guided landing of unmanned helicopters without gps sensors," in *IEEE Int. Conf. on Robotics and Aut. (ICRA)*, 2014, pp. 3096–3101.
[11] M. Schulz, F. Augugliaro, R. Ritz, and R. D'Andrea, "High-speed, steady flight with a quadrocopter in a confined environment using a tether," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1279–1284.
[12] X. Xiao, J. Dufek, and R. Murphy, "Benchmarking tether-based uav motion primitives," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2019, pp. 51–55.
[13] B. M. Rocamora, R. R. Lima, K. Samarakoon, J. Rathjen, J. N. Gross, and G. A. S. Pereira, "Oxpecker: A tethered uav for inspection of stone-mine pillars," *Drones*, vol. 7, p. 73, 1 2023.
[14] J. D. X. Xiao, Y. Fan and R. Murphy, "Indoor uav localization using a tether," in *2018 IEEE Int. Symposium on Safety, Security, and Rescue Robotics (SSRR), Philadelphia, PA, 2018*. IEEE, 2018, pp. 1–6.
[15] E. Lockwood, *A Book of Curves*, 1st ed. Cambridge University Press, ISBN: 103156, 1961.
[16] X. Xiao, J. Dufek, M. Suhail, and R. Murphy, "Motion planning for a uav with a straight or kinked tether," in *2018 IEEE/RSJ Int. Conf. on Int. Robots and Sys. (IROS), Madrid*, 2018, pp. 8486–8492.
[17] X. Xiao, J. Dufek, and R. Murphy, "Autonomous visual assistance for robot operations using a tethered uav," in *Field and Service Robotics*, 01 2021, pp. 15–29.
[18] L. Petit and A. L. Desbiens, "Tape: Tether-aware path planning for autonomous exploration of unknown 3d cavities using a tangle-compatible tethered aerial robot," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 550–10 557, 2022.
[19] S. Martínez-Rozas, D. Alejo, F. Caballero, and L. Merino, "Optimization-based trajectory planning for tethered aerial robots," in *IEEE Int. Conf. on Robotics and Aut. (ICRA)*, 2021, pp. 362–368.
[20] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
[21] P. Krüsi, P. Furgale, M. Bosse, and R. Siegwart, "Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments," *Journal of Field Robotics*, vol. 34, no. 5, pp. 940–984, 2017.
[22] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart, "Signed distance fields: A natural representation for both mapping and planning," in *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*. University of Michigan, 2016.
[23] L. Han, F. Gao, B. Zhou, and S. Shen, "FIESTA: fast incremental euclidean distance fields for online motion planning of aerial robots," in *2019 IEEE/RSJ Int. Conf. on Intel. Robots and Systems, IROS 2019, Macau, China, Nov. 3-8, 2019*. IEEE, 2019, pp. 4423–4430.
[24] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, "Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance," *IEEE Access*, vol. 6, pp. 58 383–58 394, 2018.
[25] S. Agarwal, K. Mierle, *et al.*, "Ceres solver," http://ceres-solver.org, 2021.
[26] J. A. Gallego, F. A. González, and O. Nasraoui, "Robust kernels for robust location estimation," *Neurocomputing*, vol. 429, pp. 174–186, 2021.
[27] F. Caballero and L. Merino, "DLL: Direct LIDAR Localization. A map-based localization approach for aerial robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 5491–5498.