sensors

Article

Indoor Visual Exploration with Multi-Rotor Aerial Robotic Vehicles

Panagiotis Rousseas [1], George C. Karras [1,2], Charalampos P. Bechlioulis [1,3,*] and Kostas J. Kyriakopoulos [1]

[1] Control Systems Laboratory, National Technical University of Athens, 15772 Athens, Greece; prousseas@mail.ntua.gr (P.R.); gkarras@uth.gr (G.C.K.); kkyria@mail.ntua.gr (K.J.K.)
[2] Department of Informatics and Telecommunications, University of Thessaly, 35100 Lamia, Greece
[3] Department of Electrical and Computer Engineering, University of Patras, 26504 Patras, Greece
* Correspondence: chmpechl@upatras.gr

Abstract: In this work, we develop a reactive algorithm for autonomous exploration of indoor, unknown environments for multiple autonomous multi-rotor robots. The novelty of our approach rests on a two-level control architecture comprised of an Artificial-Harmonic Potential Field (AHPF) for navigation and a low-level tracking controller. Owing to the AHPF properties, the field is provably safe while guaranteeing workspace exploration. At the same time, the low-level controller ensures safe tracking of the field through velocity commands to the drone's attitude controller, which handles the challenging non-linear dynamics. This architecture leads to a robust framework for autonomous exploration, which is extended to a multi-agent approach for collaborative navigation. The integration of approximate techniques for AHPF acquisition further improves the computational complexity of the proposed solution. The control scheme and the technical results are validated through high-fidelity simulations, where all aspects, from sensing and dynamics to control, are incorporated, demonstrating the capacity of our method in successfully tackling the multi-agent exploration task.

Keywords: unmanned aerial vehicles; multi-rotor aerial vehicles; autonomous navigation; robotic exploration; multi-agent systems

check for updates

Citation: Rousseas, P.; Karras, G.C.; Bechlioulis, C.P.; Kyriakopoulos, K.J. Indoor Visual Exploration with Multi-Rotor Aerial Robotic Vehicles. *Sensors* **2022**, *22*, 5194. https://doi.org/10.3390/s22145194

Academic Editors: Angelo Cenedese and Giulia Michieletto

Received: 12 June 2022
Accepted: 8 July 2022
Published: 11 July 2022

1. Introduction

In recent years, the field of Robotics has gone through significant advancements in both software as well as hardware. One of the most notable subsets of the field is Unmanned Aerial Vehicles (UAVs). The rise in computing power, advances in energy storage technologies and the development of sophisticated control algorithms have resulted in UAVs being widely employed in research and industry alike. The rapid adoption of such platforms is well-motivated, owing to the versatility and recent accessibility of the platforms, which renders them suitable for a wide range of applications. In this work, we aim to utilize the capabilities of a fleet of multi-rotor autonomous aerial vehicles (drones) for the exploration of indoor spaces. While a single platform can accomplish such a task, it is evident how employing multiple agents results in a decrease in the overall exploration time, as well as overcomes possible limitations due to, e.g., limited battery life. In order to successfully accomplish this task, already existing, robust and proven technologies for visual sensing of the drone's environment are implemented. The path planning aspect will be tackled through a provably-correct custom framework based on Harmonic Artificial Potential Field (AHPFs). Finally, the control framework is completed through a low-level, custom reactive field tracker, which ensures that the robots track the field safely without sacrificing the full exploration guarantees provided by the AHPF-produced velocity field. Our framework is built around the Robot Operating System (ROS) [1], which enables sensing, communication and actuation of robotic systems. Owing to the aforementioned features, our algorithm exhibits superior performance with respect to smoothness and computational complexity, while also being applicable in realistic scenarios, as validated in a high-fidelity simulation environment.

The sequel is organized as follows: Section 2 concentrates on related work to all key aspects of the proposed scheme. Section 3 formulates the treated problem mathematically, with the proposed solution being extensively discussed in Section 4. We conclude our paper with Section 6 where the limitations of our approach are highlighted along with possible solutions, followed by the Conclusions Section 7, where future prospects are also discussed.

2. Related Work

2.1. State-of-the-Art on Autonomous Robot Exploration

The problem of autonomous exploration can be summarized as formulating methods for maximizing the information gained over a specific, initially unknown area in which one or more robots operate. Search and rescue reconnaissance missions are a typical example where, due to possible hazardous environments, e.g., after an environmental disaster, robots can be employed to access remote or hazardous areas that are inaccessible to humans [2]. A common approach, termed Frontier-based exploration, consists of defining (multiple) frontiers as regions on the boundary of the so-far explored space that are free of obstacles and/or obstructions. Thus, these frontiers can be employed to provide feasible explorative paths for the robots. The latter are guided towards the aforementioned frontiers while at the same time building a map of the environment until no such frontiers remain [3,4]. There exist various tools in the literature for robots to associate the predicted information gain with the overall exploration progress such as the reduction in entropy of a Rao–Blackwellized Particle Filter [5], Shannon and Renyi entropy [6], Cauchy–Schwarz quadratic mutual information [7] or Gaussian Processes and Bayesian Optimization [8]. Some more recent efforts on autonomous exploration expand on classical navigation algorithms, such as [9], where a Rapidly Random exploring Trees (RRT)-based planner for Unmanned Aerial Vehicles (UAV) is introduced. In [10], a Topology–Grid Hybrid Map (TGHM) scheme is formulated for autonomous exploration. Modern approaches, such as Reinforcement Learning (RL), have also been employed to tackle the exploration problem [11].

It is evident that for the exploration task to be successful, safety during navigation, i.e., lack of collisions between the robots and the obstacles located within the operating area, is pivotal. Another class of solutions to the exploration problem that integrates safety into the exploration problem are Artificial Potential Field (APF) methods, which are employed to construct an underlying velocity vector field over the explored area for the autonomous guidance of the robot towards the unexplored areas. While APFs exhibit several advantages, i.e., reactivity and continuity of the method. etc., some potential fields might present local equilibria that may stall the robot in undesired configurations. To alleviate these limitations, a specific class of APF, namely Harmonic Potential Fields (HPFs), are employed owing to their property of exhibiting no local minima by design. These methods can therefore provide an efficient alternative for autonomous guidance during exploration. Towards this direction, HPFs with Dirichlet boundary conditions [12], Voronoi diagrams and distorted HPFs [13] both for 2D and even for 3D spaces [14] have been presented in the literature. However, constructing such harmonic potential fields often presents numerical and computational challenges and are especially susceptible to the curse of dimensionality, scaling poorly as the size and dimensions of the workspace increase. In order to tackle the above problems, several approaches have been developed, such as restricting the necessary relevant computations of the APF on a local window around each robot [15] or multi-grid methods [16]. Nevertheless, notably, in the aforementioned HPF-based frameworks, the APF was computed by applying finite differences, thus requiring the discretization of the entire exploration domain, which results in a significant increase in the computational complexity proportional to the grid density. More importantly, in most cases, the proposed solution is limited to two-dimensional mobile robots with disc-sector sensing (LIDAR-like sensors).

2.2. State-of-the-Art on Multi-Robot Exploration

Fixed-base station multi-agent exploration has been extensively investigated in the literature through a variety of approaches [17–21]; however, communication constraints in such centralized frameworks have not been explicitly considered. Nevertheless, since communication constraints are a persistent aspect of real-world applications, which can hinder the efficacy and implementability of any method, there has recently been a focus on treating multi-agent-related problems under such constraints. The latter have been commonly treated through the notion of connectivity, which concerns the communication ability between robots and the fixed station. Consequently, a widely employed approach consists of maintaining the connectivity between the robots and the base station, directly or in a multi-hop manner, continuously during the execution of any related task. This constraint is especially important in the context of search and rescue applications, where valuable real-time information, such as image streaming, is pivotal [22–24]. Other approaches treat communication of acquired information to the base station periodically, in a discontinuous re-connection fashion, which can be implemented either synchronously [25] or asynchronously [26]. In the latter case, periodic connectivity may not be enforced as a hard constraint; for example, in the role-based distributed strategy [27], robots are free to explore the unknown environment with no communication restrictions. In order, however, to communicate relevant information, specifically selected rendezvous points are selected, where asynchronous updates of the relevant robot maps of the environment to the base station take place via communication relays. At the same time, stronger forms of asynchronous connectivity, e.g., line-of-sight communication [28], or a small set of behavioral and message exchanges between robots and dropped beacons [29], have also been investigated. Finally, communication constraints can also be defined with the goal of ensuring global connectivity only at the deployment locations of the robots through recurrent connectivity strategies. This results in enforcing recurrent connectivity upon the collection of new data by each robot [30–32]. The motivation behind such approaches mainly rests on the fact that robots can be disconnected for arbitrarily long periods of time while establishing a connection only when new information has been collected.

2.3. Proposed Method

Within the existing paradigm, our method aims at constructing a provably correct exploration scheme that employs the capabilities of a fleet of drones to explore an indoor environment in finite time in a safe manner. In order to take advantage of the multiple drones, no explicit communication network is assumed, and each drone can communicate with others wirelessly if and when visual identification between drones is established. This decentralized approach, while not explicitly ensuring optimality during the exploration task, results in a computationally attractive and easily implementable approach in practice, as it only necessitates an adequately equipped drone fleet.

Additionally, in order to formulate a computationally attractive method for real-time octomap building, we decouple the 3D mapping aspect from the navigation task; that is, the navigation framework relies solely on a computationally efficient 2D sector-based scheme. Conversely, the on-board visual information is employed to build a 3D representation of the workspace online. This results in a more robust navigation framework, owing to its simple workspace representation that is more attractive in the context of the on-board computing power of modern multi-rotor platforms. By guaranteeing full workspace exploration, the mapping aspect is also ensured to provide reliable results.

To summarize, our contributions are:

- Development of an AHPF-based exploration algorithm for a multi-rotor platform,
- Extending the above scheme to the multi-robot exploration problem,
- Integrating the aforementioned exploration framework with a scheme for single-agent visual map-building of unknown workspaces, combined with an inter-agent information exchange aspect.

The goal is, therefore, to provide a computationally tractable solution for indoor exploration of multi-agent, multi-rotor platforms.

3. Problem Formulation

Consider a three-dimensional, simply-connected, static workspace $\mathcal{W} \in \mathbb{R}^3$. In this case, the topological property of "simple connectivity" dictates that there are no floating obstacles inside the workspace. Further consider a set of drones, $\mathcal{I} = \{1, 2, \cdots, I\}$, operating within the workspace $\mathcal{W}$, with $p_i \in \mathcal{W} - \partial\mathcal{W},\ i \in \mathcal{I}$ denoting the position of the i-th robot. The problem is thus initially formulated as follows: *Assuming that the drones start from arbitrary positions within $\mathcal{W}$, the goal is to explore the whole workspace in finite time, with no collisions with the workspace boundary $\partial\mathcal{W}$ or between the drones themselves*.

In order to formulate a solution, we begin by assuming that there exists a superscribed sphere $\mathcal{S}_i^2 \subset \mathbb{R}^{\nVDash},\ i \in \mathcal{I}$ with radius R_i for each drone, such that each drone lies entirely within the i-th sphere. Then, let $\mathcal{W}_{infl}$ denote the inflated workspace, i.e.:

$$\partial\mathcal{W}_{infl} = \{q | q = z + R_{max} n(z),\ \forall z \in \partial\mathcal{W}\}, \tag{1}$$

where $R_{max} = \max\{R_i\}, i \in \mathcal{I}$ denotes the maximum radius of the set of superscribed spheres, and $n(z) : \partial\mathcal{W} \mapsto \mathbb{R}^3$ denotes the inwards-pointing unitary vector at the point z on the boundary of the workspace. Notice that by inflating the workspace, the problem is essentially transformed into the navigation of a *point robot* (this step can be easily incorporated into the proposed framework through computationally inexpensive methods). A necessary assumption that follows is that the inflation of the workspace boundary does not hinder the connectivity of the workspace; when taking the maximal radius for each drone's representation of the workspace (for compatibility purposes during map matching, as it will become apparent in the sequel, a large enough drone might not "fit" through narrow corridors of the inflated workspace. Therefore, we henceforth assume that $\mathcal{W}_{infl}$ remains **fully connected**. Nevertheless, this assumption is not that strong, as the existing platforms relevant to indoor applications are sufficiently small. This assumption can also be lifted if all drones are exactly similar, which is highly likely in practice. Further assume that each drone is equipped with a sensor that provides boundary information of the surrounding workspace, i.e.,

$$\mathcal{S}(p) = \Bigg\{ q \in \mathcal{W} : (\mathcal{B}(p, r) \wedge \mathcal{L}(p, q)) \subseteq \mathcal{W} \Bigg\} \tag{2}$$

where $\mathcal{B}(p, r)$ is the disk with radius r centered at p and $\mathcal{L}(p, q)$ is the linear segment connecting p and q inclusively. Additionally, we denote by $\mathcal{P}_i(t_s, t_e), i \in \mathcal{I}$ the i-th drone's path from time t_s until t_e. The explored region along the path $\mathcal{P}_i, i \in \mathcal{I}$ is thus defined as:

$$\mathcal{E}_i \triangleq \mathcal{E}(\mathcal{P}_i) = \cup_{p \in \mathcal{P}_i} \mathcal{S}(p)\ , i \in \mathcal{I}, \tag{3}$$

whose boundary is denoted by $\partial\mathcal{E}(\mathcal{P}_i) = \partial\mathcal{E}_F(\mathcal{P}_i) \cup \partial\mathcal{E}_O(\mathcal{P}_i)$, where $\partial\mathcal{E}_F(\mathcal{P}_i) \subset \mathcal{W} - \partial\mathcal{W}$ denotes the free part and $\partial\mathcal{E}_O(\mathcal{P}_i) \subseteq \partial\mathcal{W}$ denotes the obstacle-occupied part of the discovered boundary for the i-th drone. We additionally define an appropriate parametrization of the explored boundary, i.e., $\sigma_i : [0, 1] \mapsto \partial\mathcal{E}_i,\ i \in \mathcal{I}$.

Proposed Sub-Problems

In order to tackle the above problem, we break it down into three sub-problems, namely:

1. Localization and Mapping,
2. Path Planning,
3. Tracking Problem.

These sub-problems can now be addressed separately. Briefly, the Localization and Mapping problem essentially consists of obtaining an accurate representation of the discovered boundary $\partial\mathcal{E}_i, i \in \mathcal{I}$ and of the position $p_i,\ i \in \mathcal{I}$ of each robot. Path Planning is addressed in the context of AHPFs, where a vector field suitable for navigation is constructed. If a robot follows the field exactly, then safe exploration can be guaranteed. However, a robot's dynamics, along with possible kinematic non-holonomic constraints, prohibit exact tracking of the field. Therefore, we treat the tracking problem separately through the formulation of a provably safe, reactive tracking controller.

4. Materials and Methods

We begin with an overview of the method. Based upon the existing attitude controllers of multirotor platforms (e.g., the open source ArduPilot software https://ardupilot.org/, accessed on 1 June 2022), we formulated a two-level controller. The latter is based on a high-level navigation field for exploration [33] and a low-level tracking controller based on the unicycle kinematic model [34]. Finally, inter-agent collision avoidance is guaranteed through the manual addition of the visible subset of agents to each agent's occupancy grid. Our method provides guarantees of safety and full-workspace exploration in finite time owing to previous works [33,35].

4.1. Multirotor Kinematics and Dynamics

In this subsection, we introduce the widely employed kinematic and dynamic models of multirotor platforms. Consider the exemplary multirotor robot, as depicted in Figure 1.

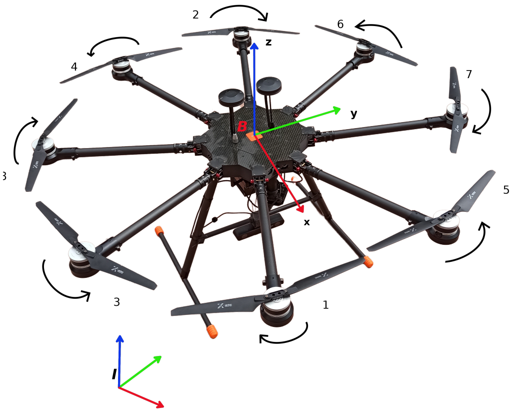


Figure 1. Multirotor's frames.

Let $\mathbf{B} = \left\{ e_{B_x} \quad e_{B_y} \quad e_{B_z} \right\}$ denote the body-fixed frame, whose origin coincides with the vehicle's center of mass. Furthermore, we define a fixed inertial frame $\mathbf{I} = \left\{ e_{I_x} \quad e_{I_y} \quad e_{I_z} \right\}$, as depicted in Figure 1. The translational and rotational dynamics of the vehicle are described by the Newton–Euler equations [36,37]:

$$\dot{\mathbf{p}} = {}^{I}\mathbf{v} = {}^{I}\mathbf{R}_B {}^{B}\mathbf{v} \tag{4}$$

$$m^{I}\dot{\mathbf{v}} = {}^{I}\mathbf{R}_B\mathbf{F} \tag{5}$$

$$\mathbf{J}\dot{\omega} = \mathbf{M} \tag{6}$$

where $\mathbf{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$, ${}^{I}\mathbf{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T$ denote the position and the linear velocity of the drone with respect to the inertial frame $\mathbf{I}$, ${}^{B}\mathbf{v} = \begin{bmatrix} u & v & w \end{bmatrix}^T$ is the linear velocity vector with respect to the body frame $\mathbf{B}$, m denotes the mass, ${}^{I}\mathbf{R}_B$ denotes the rotation matrix used to perform rotations from $\mathbf{B}$ to $\mathbf{I}$, $\mathbf{J}$ denotes the inertia matrix, and $\boldsymbol{\omega}$ denotes the angular velocity of the robot with respect to the body frame $\mathbf{B}$.

The external forces and torques that are applied to the vehicle are split into the following terms:

$$\mathbf{F} = \mathbf{F}_M + \mathbf{F}_d + \mathbf{F}_g \tag{7}$$

$$\mathbf{M} = \mathbf{M}_M + \mathbf{M}_d \tag{8}$$

where:

- $\mathbf{F_d} = C_d{}^B\mathbf{R}_I \|{}^I\mathbf{v}\|{}^I\mathbf{v}$ denotes the drag forces and C_d the drag coefficient matrix;
- $\mathbf{F_g} = m{}^B\mathbf{R}_I\begin{bmatrix}0 & 0 & -g\end{bmatrix}^T$ denotes the gravitational force, where g is the gravitational constant;
- $\mathbf{F_M} = \begin{bmatrix}0 & 0 & T\end{bmatrix}^T$ denotes the total thrust generated by the motors;
- $\mathbf{M_M} = \begin{bmatrix}\tau_x & \tau_y & \tau_z\end{bmatrix}^T$ denotes the torque produced by the motors;
- $\mathbf{M_d} = C_m\|\boldsymbol{\omega}\|\boldsymbol{\omega}$ are the drag moments with C_m denoting the drag moment coefficient matrix;

The total input thrust and moment applied to the drone depends heavily on the vehicle's geometrical characteristics and specifically on the number N of motors and the airframe configuration. In line with momentum theory, the thrust force T_i and the drag moment τ_i that are produced by the drone's propellers are assumed to be proportional to the square of the rotor's angular velocity, i.e.,

$$T_i = C_T{\omega_i}^2 \tag{9}$$

$$\tau_i = C_\tau{\omega_i}^2 \tag{10}$$

where $i = 1, .., N$ and C_T, C_τ denote the thrust and drag coefficients correspondingly.

For the quadrotor, used in simulation scenarios, the total thrust and moments are computed by the following control allocation matrix:

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} C_T & C_T & C_T & C_T \\ -C_Tl_x & C_Tl_x & C_Tl_x & -C_Tl_x \\ C_Tl_y & -C_Tl_y & C_Tl_y & -C_Tl_y \\ C_\tau & C_\tau & -C_\tau & -C_\tau \end{bmatrix} \begin{bmatrix} {\omega_1}^2 \\ {\omega_2}^2 \\ {\omega_3}^2 \\ {\omega_4}^2 \end{bmatrix} \tag{11}$$

with l_x, l_y denoting the distance of each motor with respect to the center of mass.

4.2. Autopilot and On-Board Sensors

Modern multirotor platforms operate through the application of a low-level controller, which is a cascaded PID control structure consisting of an outer position loop and an inner attitude one. More specifically, the outer position loop is responsible for converting the reference position $\mathbf{p}_d$, velocity ${}^I\mathbf{v}_d$ (or ${}^B\mathbf{v}_d$) and heading ψ_d of the vehicle to target orientation (roll ϕ_d, pitch θ_d, yaw ψ_d) and throttle. The inner attitude controller then translates the aforementioned orientation and throttle commands to motor Pulse Width Modulation (PWM) values. The state feedback is achieved by fusing sensor measurements, such as data by GPS, compass and IMU, using an Extended Kalman Filter. An overview of the control architecture is shown in Figure 2.

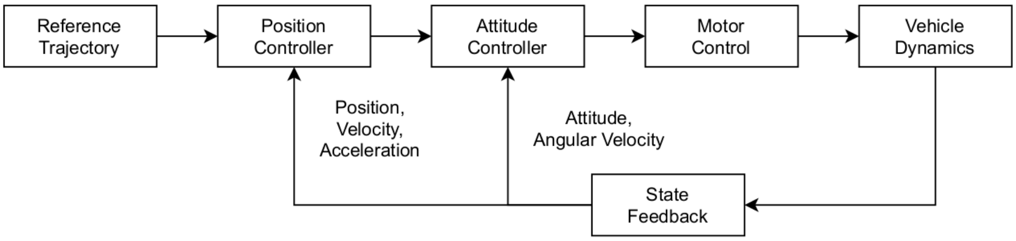


Figure 2. Ardupilot control architecture.

4.3. Localization and Mapping

In order to perform Localization and Mapping, robust existing SLAM tools will be employed. More specifically, we will employ the Hector SLAM method, as presented in [38]. This framework can be readily integrated through the available, open-source ROS package and provides odometry data for an estimate of a robot's position, along with a built-in Occupancy Grid Map (OCG), both of which are utilized for path planning and feedback control of the drones. Briefly, SLAM methods utilize sensor information (in Hector SLAM's case, a LIDAR sensor), which is processed in order to extract an estimate of the position of the sensor through matching sequentially acquired measurements. OCGs, on the other hand, consist of a 2D representation of a workspace and take the form of a matrix. Given an inertial frame of reference (e.g., the drone's take-off position), the surrounding (unknown) space is decomposed into squares (cells). The position (index pair) of each component of the aforementioned matrix corresponds to a cell inside the workspace, while the value of each component represents the probability that the respective cell is occupied by an obstacle, given the currently available information to the drone. Therefore, having such a representation, obtaining the sensed boundary $\partial\mathcal{E}$ can be directly tackled through well-known boundary extraction imaging techniques [39,40], applied over the OCG.

4.4. Path Planning

4.4.1. Velocity Field

In this section, we treat the path planning problem for a point-robot (as discussed in Section 3). In order to achieve provably complete path planning for each robot, we employ the AHPF-based method presented in [33]. Briefly, the method designs a reactive vector field for the single integrator dynamics, i.e., given the problem formulation of Section 3, along with the following dynamics:

$$\dot{p}_i = u_i,\ p_i \in \mathcal{W},\ i \in \mathcal{I}, \tag{12}$$

the authors in [33] provide a reactive vector field, which is modified herein to fit the multi-agent framework:

$$u_i = u_i(p_i) = -K_u^i S_{R_1}\Big(d(p_i;\mathcal{E}_i;p_{\mathcal{I}}^i)\Big)\nabla_p\phi(p_i;\hat{k}_i),\ i \in \mathcal{I}, \tag{13}$$

where $K_u \in \mathbb{R}$ is a constant gain, the set:

$$p_{\mathcal{I}}^i = \{p_j | j \subset \mathcal{I} - \{i\} \text{ and } p_j \in \mathcal{S}(p_i)\}, \tag{14}$$

for the drone i denotes all other drones that lie inside its sensor radius, $S_{R_1} : \mathbb{R} \mapsto \mathbb{R}$ is a scaling function to ensure safety and will be defined in the sequel, and finally, $d(p_i;\mathcal{E}_i;p_{\mathcal{I}}^i)$ is a distance function:

$$d(p_i;\mathcal{E}_i;p_{\mathcal{I}}^i) = \min\left\{\min_{z\in\partial\mathcal{E}}\{\|p_i - z\|\}, \min_{z\in p_{\mathcal{I}}^i}\{\|p_i - z\| - D_{min}^i\}\right\}, \tag{15}$$

which outputs the minimum distance between the already sensed boundary by the i-th drone and its distance from the closest of the other drones. The quantity $D_{min}^i > 0$ is a parameter that determines the least permissible distance between two drones and is chosen as $D_{min}^i < R_i,\ i \in \mathcal{I}$. Furthermore, we define the function:

$$S_a(\star) = \begin{cases} 0 & ,\star < 0 \\ 3\left(\frac{\star}{a}\right)^2 - 2\left(\frac{\star}{a}\right)^3 & ,0 \leq \star \leq a \\ 1 & ,\star > a \end{cases} \tag{16}$$

The velocity control law (13) is further derived from the gradient of the reference potential $\phi(p;\hat{k}) : \mathcal{W} \mapsto \mathbb{R}$, which is determined by a set of parameters $\hat{k}$, which will be defined in the following sub-section. The potential $\phi_i,\ i \in \mathcal{I}$ of the i-th drone is designed to satisfy the Laplace equation:

$$\nabla_p^2\phi_i(p_i;\hat{k}_i) = 0, \forall p_i \in \mathcal{W},\ i \in \mathcal{I}, \tag{17}$$

with Neumann boundary conditions:

$$\frac{\partial \phi_i}{\partial p} = k\left(\sigma_i^{-1}(q), t\right),\ \forall q \in \partial \mathcal{E}_i,\ i \in \mathcal{I}, \tag{18}$$

where $\sigma^{-1}(q)$ is the inverse function of the parametrization of the boundary introduced in Section 3. Effectively, the function $k(\sigma^{-1}(q), t)$ dictates whether a point on the boundary $q \in \partial\mathcal{E}$ is attractive $(k < 0)$ or repulsive $(k > 0)$. In order for the Neumann boundary problem to admit a solution, the boundary conditions should satisfy the compatibility constraint:

$$\int_{\partial \mathcal{E}} k \mathrm{d}s = 0. \tag{19}$$

We, henceforth, refer to such a set of boundary conditions as *compatible*. The reason why we employ such a potential for navigation is that such solutions lack local equilibria inside the domain over which (17) is solved. This property holds according to the minimum–maximum principle, which dictates that solutions to the Laplace equation exhibit minimal and maximal values only at the boundary of their domain. The boundary condition (18) further ensures that the resulting gradient vector field is safe, i.e., the resulting velocities point inwards at the boundary.

4.4.2. Boundary Discretization

It is evident that the Laplace problem (17), along with the boundary conditions (18), necessitates the definition of the function $k(\cdot)$, which would result in a problem with an infinite-dimensional boundary condition, as (18) is defined over a continuous set. In order to handle the boundary conditions numerically, we employ the following parametrization of k over a **finite** number of control points $\hat{k}$, i.e.,

$$k_i \triangleq k\left(\sigma_i^{-1}(q), t, \hat{k}_i\right),\ \hat{k}_i = \left[k_i^1, \cdots, k_i^{N_{C_p}}\right]^{\mathrm{T}},\ N_{C_p} \in \mathbb{N},\ i \in \mathcal{I}, \tag{20}$$

where each $k_i^j, j \in \{1, \cdots, N_{C_p}\}$ corresponds to points q_j placed on the observed boundary. For any boundary point q_j, $\hat{k}_i^j$ corresponds to the exact value of the boundary condition (18) for the i-th drone, while the same value for any point $q \neq q_j$, $k\left(\sigma_i^{-1}(q), t, \hat{k}_i\right)$ is chosen as a linear combination of nearby control point values. The discretization scheme is illustrated in Figure 3. In addition, we equip the specified boundary conditions $\hat{k}$ with appropriately designed adaptive laws, such that no additional (locally) stable equilibria appear in $\mathrm{int}(\partial\mathcal{E})$.

These values are tuned according to the adaptive law:

$$\dot{\hat{k}} = c\mu(\hat{k}_t - \hat{k}) + b_e(\hat{k}_t - \hat{k}), \tag{21}$$

(the index i corresponding to each drone is omitted for the sake of clarity) where $c, \mu, b_e, \hat{k}_t$ are provided in [33] and are omitted here for the sake of brevity. Briefly, the values $\hat{k}_t$ operate as reference values for the boundary conditions. It is evident that over time, (21) results in $\hat{k}$ converging to the reference values. Additionally, c renders the first term of (21) zero when the robot is near a critical point of the field ϕ while b_e ensures that the trajectories of the robot remain safe for all time. Finally, μ is employed to modify the convergence rate.

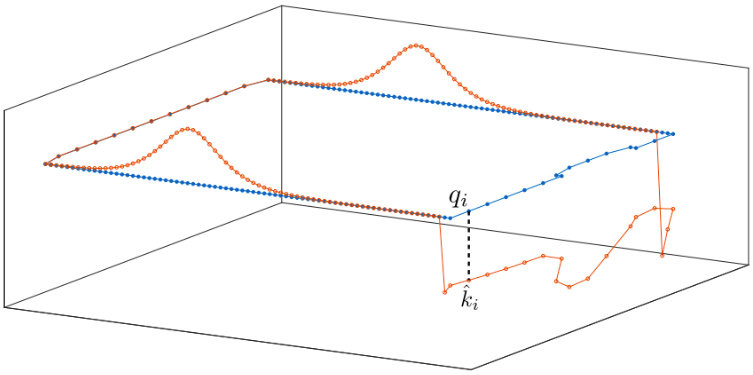


Figure 3. The discretization of boundary values using the control points $\hat{k}_i$ on the boundary points q_i. The blue line and dots represent $\partial\mathcal{E}$ and the boundary points q_i, respectively. The red line illustrates k, and the control points $\hat{k}_i$ are shown as red circles.

4.4.3. Fast Multipole Boundary Elements

In order to obtain a solution to the Laplace Equation (17) given the boundary conditions (18) dictated by the function k of the preceding sub-section, we employ a Boundary Element Method (BEM) and, more specifically, the Fast Multipole Boundary Elements Method (FBEM), as employed in [33]. Briefly, in FMBEM, the computational cost of the conventional BEM (which is $O(n^2)$, where n is the number of boundary elements used for approximating the domain's boundary) is significantly reduced through the computation of approximate solutions for the corresponding boundary value problem, with a specified error. In order to accomplish this, a quadtree decomposition is employed in order to obtain a hierarchical subdivision of the elements that describe the boundary of the solution domain. Subsequently, multipole expansion is employed to approximate the interaction among elements (with respect to the Laplace equation). Therefore, an analytical computation of every pairwise element interaction is avoided, at the expense, however, of an approximation error, which is nevertheless bounded. This process results in a reduction in the computational complexity down to $O(n)$. Finally, in FMBEM, there is no need to store the dense and non-symmetric matrix that is part of the conventional BEM if an iterative solver is employed, e.g., GMRES [41], which consequently also results in $O(n)$ memory requirements.

4.4.4. A Brief Discussion about the Algorithm

In this subsection, we briefly explain how the FBEM solution to the problems (17) and (18) is employed along with (21) to provides a safe navigation field suitable for exploration. As in [33], a solution at time $T > 0$ to the problems (17) and (18) is obtained by applying the FBEM method, along with the boundary conditions that stem from the parameters' values at that time, i.e., $\hat{k}(T)$. Once such a solution is obtained, the linear dependence of the potential on these parameters, along with the adaptive control law (21), provide a safe field for exploration, as proven by the technical results in [33]. As each drone explores the workspace, the Laplace problem is updated according to newly-discovered parts of the boundary, and successive solutions are obtained. In the following sub-section, we provide the technical results of [33] that prove the asserted claims.

4.4.5. Technical Results

In this subsection, we include some of the technical results presented in [33] for completeness while omitting the relevant proofs for brevity. For details, we direct the reader to [33].

Proposition 1. *Assuming that $\hat{k}$ is compatible at $t = 0$, the adaptive law* (21) *guarantees that $\hat{k}$ will remain compatible for all time.*

Proposition 2. *If the potential $\phi(p, \hat{k})$ is unsafe, the adaptive law* (21) *guarantees that it will become safe in a finite amount of time.*

Proposition 3. *All equilibria of the dynamic system:*

$$\dot{z}_i = \left[\dot{p}_i^T, \dot{\hat{k}}_i^T\right]^T \tag{22}$$

located in $int(\mathcal{E})$ are unstable.

Proposition 4 (Proposition (8) in [33])**.** *The proposed control protocol* (13) *and* (21) *ensures that the trajectories of $\dot{p}_i = u_i$, $i \in \mathcal{I}$ are safe.*

The above propositions ensure that the proposed velocity field has the necessary properties that will result in safe exploration of the whole workspace for each drone in finite time. Additionally, in (13), we propose a modification with respect to [33], which includes the distances between drones that lie inside the sensor radius of each other and results in safety between drones. In order to guarantee this additional type of safety, an additional feature is proposed; namely, within the OCG of each drone, we place a "fictional" obstacle, equivalent to the superscribing sphere $\mathcal{S}_i^2$ that was discussed in Section 3.

4.5. Tracking Controller

While we have introduced a provably complete navigation field for each drone, we need to ensure that, despite the inability of a drone to follow the single integrator dynamics exactly (12), safety is not violated. We assume that there exists an attitude controller over the drone dynamics that takes reference velocities as inputs and results in an adequate response by the drone. Such controllers are in widespread use in the industry and are standard in any available drone platform [42,43]. Additionally, we need to ensure that the drone's visual system is also pointing towards the direction of the motion, which, besides ensuring proper sensing, results in a better response with respect to the commanded velocities, as the drone is not commanded to alter its direction of motion in a sharp, non-smooth fashion. If the drone's low-level tracking controller was commanded to change its direction in such a fashion, this would result in a delayed response (owing to the drone's dynamics and its momentum), which might render its motion unsafe. To this end, we first introduce the yaw angle of the drone $\theta \in \mathcal{S}^1$ (where $\mathcal{S}^1$ is the set of 1-dimensional rotations), which can be obtained with respect to an inertial frame of reference $\{I\}$ as $\theta = \text{atan2}({}^I p_x, {}^I p_y)$, where $[{}^I p_x, {}^I p_y]$ denotes the x and y axes of the drone's own frame of reference expressed in $\{I\}$. Then, we propose the following reactive control law:

$$\dot{\theta} = -K_\theta(1 - \cos(\alpha))\sin(\alpha), \tag{23}$$

where α is the relative error angle, i.e.,

$$\alpha = \theta - \theta_{\text{ref}}, \tag{24}$$

where θ_{ref} is the reference angle of the underlying velocity field, i.e.,

$$\theta_{\text{ref}} = \text{atan2}({}^I u_x(p), {}^I u_y(p)),\; u(p) = \left[{}^I u_x(p), {}^I u_y(p)\right]^{\text{T}}. \tag{25}$$

We prove how this control law stabilizes the reference angle at $\alpha = 0 \Leftrightarrow \theta = \theta_{\text{ref}}$:

Proposition 5. *The controller* (23) *results in zero relative error angle over time for a constant reference angle θ_{ref}.*

Proof. Consider the Lyapunov Candidate function $V(\alpha) = (1 - \cos(\alpha))^2$. This funtion is evidently positive definite $\forall \alpha \in \mathcal{S}^1$, with

$$V(\alpha) = 0 \iff \alpha = 0 \iff \theta = \theta_{\text{ref}}. \tag{26}$$

Additionally, its derivative with respect to time is:

$$\begin{aligned} \dot{V} &= +2(1 - \cos(\alpha))\sin(\alpha)(\dot{\theta} - \dot{\theta}_{\text{ref}}) \\ &= -2K_\theta(1 - \cos(\alpha))^2\sin^2(\alpha) - 2(1 - \cos(\alpha))\sin(\alpha)\dot{\theta}_{\text{ref}} \end{aligned} \tag{27}$$

which for $\dot{\theta}_{\text{ref}} = 0$ is negative for $\alpha \neq 0$, thus concluding the proof. □

However, it is evident that $\dot{\theta}_{\text{ref}} \neq 0$ as the drone moves inside the workspace, which changes the vector of the reference velocity and, therefore, its reference angle as well. To remedy this, we propose the following velocity controller:

$$u_{\text{ref},i}(p) = K_u\left[\gamma\sigma_p\left(\frac{d(p_i;\mathcal{E}_i;p^i_{\mathcal{J}})}{\epsilon_p}\right) + (1-\gamma)\sigma_u\left(\frac{\langle\hat{e}^i_n, u(p_i)\rangle(-\langle\hat{e}^i_n, \hat{n}(p_i)\rangle)}{\epsilon_u + |\langle-\hat{e}^i_n, u(p_i)\rangle||\langle\hat{e}^i_n, \hat{n}\rangle|}\right)\right]u(p_i), \tag{28}$$

where $\gamma \in (0,1)$, $\sigma_p, \sigma_u : \mathbb{R}^+ \mapsto [0,1]$ are smooth and monotonic bump functions such that $\sigma_p(0) = 0, \sigma_p(\infty) = 1$. The quantity $\hat{e}^i_n$ denotes the unitary vector along the direction of the i-th drone's heading (forward-looking direction), $\hat{n}(p)$ denotes the unitary, inwards-pointing vector at the closest point of boundary to the drone's position p, and $\langle\cdot,\cdot\rangle$ denotes the classical inner product between two vectors, while $u(p)$ denotes the reference velocity field presented in Section 4.4. More specifically, we define the normal vector at the i-th drone point p_i:

$$\hat{n}(p_i) = n(z(p_i)),\ z(p_i) = \arg\min_{z\in\partial\mathcal{E}_i\cup p^i_{\mathcal{J}}}\{\|p_i - z\|\}, \tag{29}$$

where evidently $n(z),\ z \in \partial\mathcal{E}_i \cup p^i_{\mathcal{J}}$ denotes the normal vector to the boundary of the so-far explored region or the closest of the rest of the robots (if it lies within the i-th robot's sensor radius). The first term in (28) becomes zero as the drone approaches the boundary of another drone (where $d(p;\mathcal{E}_i;p^i_{\mathcal{J}}) \to 0$). The second term approaches zero as:

- The vectors $\hat{e}_n, u(p)$ point to different directions, and
- The vectors $\hat{e}_n, \hat{n}$ point to different directions.

The first condition means that the velocity of the drone decreases if the robot is misaligned with respect to the velocity field, while the second condition decreases the velocity if the drone is pointing outwards at the boundary. Additionally, this implies that if the functions σ_p, σ_u, along with the tuning parameters $\gamma, \epsilon_p, \epsilon_u$, are chosen appropriately, they provide an adequate bound to $\dot{\theta}$, which renders the controller (23) asymptotically stable for $\dot{\theta}_{\text{ref}} \neq 0$. This results in safe and successful navigation, regardless of the drone's dynamics, if the drone's own attitude controller can perform adequately. The efficacy of the above-proposed control protocol in successfully completing the multi-drone exploration task is demonstrated in rigorous, high-fidelity software-in-the-loop (SITL) simulations.

4.6. Octopmap Building

Since the navigation aspect is fully addressed through a two-dimensional OCG framework, the visual information is employed independently in order to build a three-dimensional representation of the indoor environment. This is achieved through integrating already existing, robust tools for 3D mapping. The independence of the navigation and 3D mapping modules results in a computationally more attractive scheme. We demonstrate the efficacy of the proposed method in building a complete map in the results section of our work in a realistic, synthetic simulation environment.

4.7. Drone Communication and Map Merging

In order to exploit the full potential of the multi-drone exploration framework, a method for establishing communication between the drones to exchange the robots' maps needs to be formulated. Subsequently, each robot should process the exchanged map information and merge the latter with its own map in order to take advantage of the exploration of the rest of the drone fleet. To accomplish this, we formulate a decentralized, aperiodic communication strategy. In order to provide a simple solution that negates the need for a centralized station, we assume that during the exploration process, a number of $C \in \mathbb{N}$ drones, denoted by $\{i_1, i_2, \cdots, i_C\} \triangleq \mathcal{C} \subset \mathcal{I}$, lie in the visual field of one or more of the aforementioned drones. We formulate the communication scheme for one drone, denoted by $i \in \mathcal{I}$, and employ the same strategy for any other drone. In order to verify the presence of the set of drones $\mathcal{C}_i \subset \mathcal{I} - \{i\}$ for the i-th drone within its vicinity, we use visual means. This choice further motivates the Lyapunov-based controller for each drone's heading angle θ (23), as it is necessary that each drone's visual system is appropriately oriented towards its direction of motion in order to possibly detect any other neighboring drone. We further assume that the presence of two drones within the field of view of any of the latter implies that wireless communication is feasible. Having successfully satisfied the above conditions, the i-th drone can send its formed OCG map towards every drone $j \in \mathcal{C}_i$ in its vicinity.

A pivotal aspect to the success of the scheme is the merging, by the i-th drone, of a composed map from the set of neighboring drones $\mathcal{C}_i$. There exist several methods in the literature that achieve this merging, from [44], where SLAM was extended for a set of multiple robots based on an Extended Kalman Filter (EKF), to [45], which employed a particle filter to achieve multi-robot SLAM. In another approach [46], the author proposes a hybrid method for the formation of a maximum likelihood map along with a Monte-Carlo position estimator. Furthermore, there are off-the-shelf ROS packages available for map-merging, e.g., [47]. In this work, we employ this tool to achieve the map merging for each drone.

5. Results

In order to demonstrate the efficacy of the proposed scheme, we have constructed a synthetic simulation scenario where two drones operate within an a priori unknown indoor environment, comprising of planar vertical walls. The environment is depicted in Figure 4. As the drones navigate autonomously, they gather visual data and build an Octomap 3D representation of their environment. Successive snapshots during the exploration process are depicted in Figure 5. We note that both Octomaps, one formed by each drone, are merged and presented in the respective snapshots for illustration purposes. The final Octomap, along with the merged OCG maps after the exploration process has been completed, are presented in Figure 6. Furthermore, the trajectories of the two drones overlayed on top of the final OCG map are depicted in Figure 7. Finally, a video of the experiment can be accessed through the following hyperlink: https://youtu.be/yDP_Zz-P0lE (accessed on 11 June 2022).

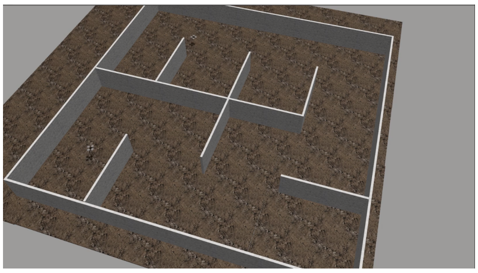


Figure 4. The synthetic simulation environment along with the two autonomous drones.

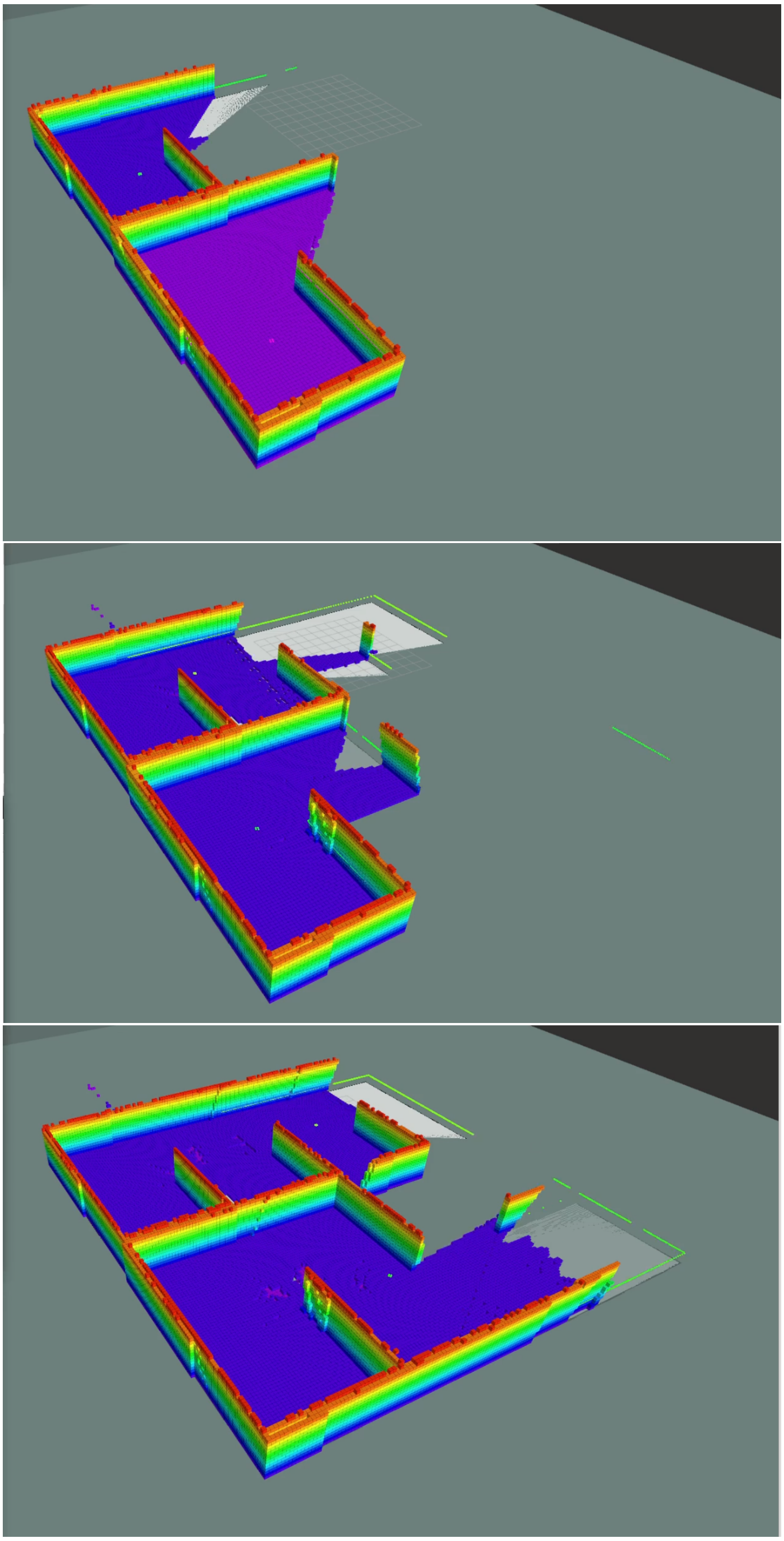
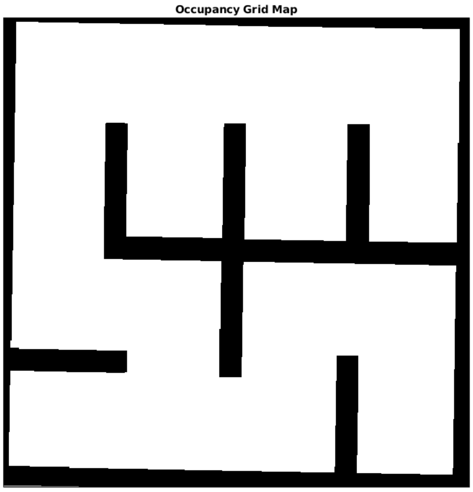


Figure 5. Snapshots of the map-building process during exploration.



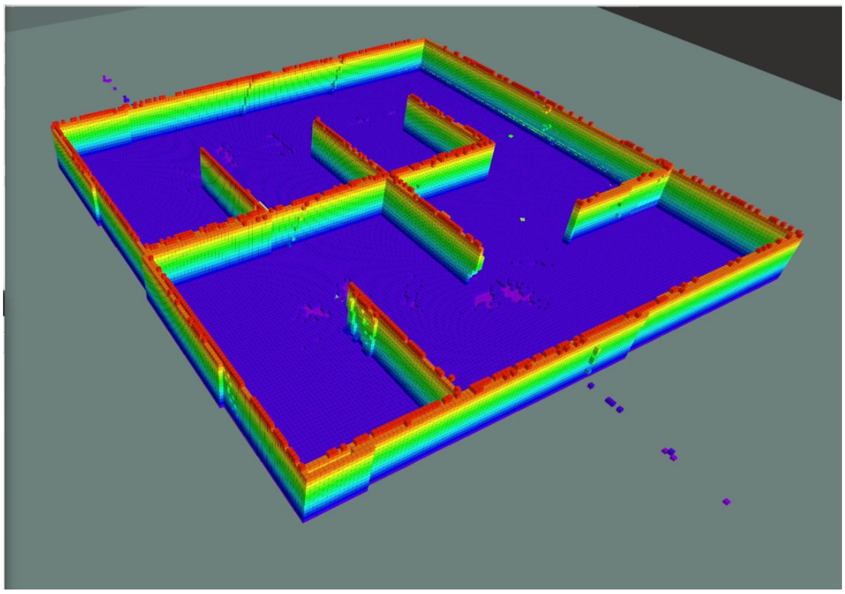


Figure 6. Final results of the exploration/mapping framework. The complete 2D OCG map of the environment (**top**), along with the final 3D Octomap representation of the latter (**bottom**).

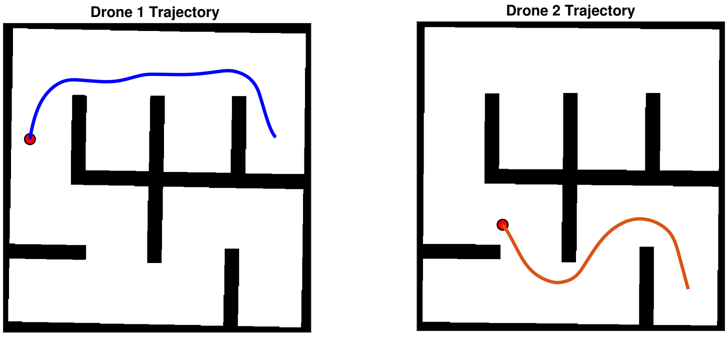


Figure 7. The trajectories that the two multirotor robots executed during the exploration process.

6. Limitations

The main limitations of this work are related to sensing and tuning. Most multi-rotor platforms are not equipped with an omni-directional Lidar, but rather their sensing is limited over a smaller subset of the full 360° range. This specification might result in the formal assumptions of the method being violated, owing to the drone "exiting" the sensed area over which the AHPF is defined, evidently due to the drone's blindspots. This issue can be addressed if, prior to initiating exploration, the drone performs a full rotation about the vertical axis of its body frame of reference, thus forming an appropriate "sensed region" around itself and negating the probability of exiting the aforementioned area.

The second limitation is due to the use of Lyapunov/PID tracking controllers. Since these controllers are model-agnostic, some tuning is required to ensure proper tracking of the underlying AHPF. Nevertheless, tuning the respective parameters did not prove to be difficult in practice, especially since a conservative choice of maximum linear velocity can easily result in safe navigation, however, sacrificing total exploration time. An interesting future direction rests on exploring tuning methods for the parameters of the herein proposed controllers.

Another envisioned limitation—although not observed in practice during the evaluation of the proposed scheme—lies in the computational complexity of the method in large workspaces. While the FBEM is advantageous insofar as it scales linearly with respect to the size of the boundary (in contrast to scaling proportionally to the area of the workspace), very large workspaces could still pose an issue as the whole workspace is considered during the solution to the Laplace problem. This could evidently be sidestepped, as parts of the explored workspace could be discarded. This is another interesting future research area where we intend to formulate a polygonal convex decomposition method for reducing the computational load of the proposed scheme in large workspaces.

A final limitation stems from the limited battery life of drones, which is crucial in the context of exploration, as the size of the workspace is not known a priori. This could be addressed through existing approaches, such as [48].

7. Conclusions

In this work, we have formulated a method for solving the cooperative exploration task through a multirotor-based scheme. The proposed framework relies on existing robust tools for mapping and localization, while its novelty lies in the application of a Harmonic Potential-based exploration algorithm. Through the technical results, along with the in-simulation validation of the proposed control scheme, our method effectively tackles the multi-agent exploration problem with multi-rotor aerial robots. We, therefore, conclude that the herein posed research problem was successfully solved. Furthermore, our method exhibits some significant advantages. Since the proposed scheme employs the FBEM method, transforming the problem into a boundary problem on a 1D curve, it scales favorably with respect to the size of the workspace ($\mathcal{O}(n)$ complexity). This is in contrast to other popular methods, such as other popular planners for navigation in unknown workspaces, such as A* [49], which, at best, scales polynomially or, at worst, exponentially with respect to the size of the workspace. Another advantage of the method is the two-layer approach through a tracking controller over a navigation field. This results in a more robust scheme with respect to the drone's dynamic behavior, as the internal attitude controller of the multi-rotor platform is employed to tackle the stabilization of its highly non-linear dynamics. Furthermore, our method is modular, which means that it can be easily employed in several robotic platforms (e.g., mobile robots) with minor modifications. This results in a robust and computationally efficient framework for cooperative exploration of indoor areas, as demonstrated in a high-fidelity realistic simulation scenario.

With regards to future work, we intend to improve several aspects of the algorithm. First of all, the parameters of the proposed tracking controller are tuned by hand. It would, however, be possible to formulate tuning laws based on some optimality criterion, e.g., exploration time, explored area coverage, etc., in a multi-agent, game-theoretic approach.

Additionally, since computing the field for the whole workspace is unnecessary in several cases, thus adding to the computational complexity of the method with no apparent advantages, we intend on formulating an algorithm for disregarding subsets of the explored workspace that do not "contribute" to the discovery of new parts of the workspace, based on polygonal decomposition. This will effectively bound the number of parameters as in [50].

Author Contributions: Conceptualization, G.C.K., C.P.B. and K.J.K.; methodology P.R., C.P.B. and G.C.K.; software, P.R. and G.C.K.; validation, P.R. and G.C.K.; formal analysis P.R., writing—original draft preparation P.R.; writing—review and editing G.C.K., C.P.B. and K.J.K.; supervision K.J.K.; project administration, K.J.K.; funding acquisition, K.J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded by the European Union's Horizon 2020 Research and Innovation Program PATHOCERT—Pathogen Contamination Emergency Response Technologies under grant agreement No. 883484.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Stanford Artificial Intelligence Laboratory. *Robotic Operating System*; Stanford Artificial Intelligence Laboratory: Stanford, CA, USA, 2022.
2. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*; The MIT Press: Cambridge, MA, USA, 2005.
3. Yamauchi, B. A frontier-based approach for autonomous exploration. In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. Towards New Computational Principles for Robotics and Automation, Monterey, CA, USA, 10–11 July 1997; pp. 146–151. [CrossRef]
4. Lu, L.; Redondo, C.; Campoy, P. Optimal Frontier-Based Autonomous Exploration in Unconstructed Environment Using RGB-D Sensor. *Sensors* **2020**, *20*, 6507. [CrossRef] [PubMed]
5. Stachniss, C.; Grisetti, G.; Burgard, W. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In Proceeding of the Robotics: Science and Systems I, Cambridge, MA, USA, 8–11 June 2005; The MIT Press: Cambridge, MA, USA, 2005; pp. 65–72. [CrossRef]
6. Carrillo, H.; Dames, P.; Kumar, V.; Castellanos, J.A. Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and Rényi Entropy. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 487–494. [CrossRef]
7. Charrow, B.; Liu, S.; Kumar, V.; Michael, N. Information-theoretic mapping using Cauchy-Schwarz Quadratic Mutual Information. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 4791–4798. [CrossRef]
8. Bai, S.; Wang, J.; Chen, F.; Englot, B. Information-theoretic exploration with Bayesian optimization. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 1816–1822. [CrossRef]
9. Zhao, L.; Yan, L.; Hu, X.; Yuan, J.; Liu, Z. Efficient and High Path Quality Autonomous Exploration and Trajectory Planning of UAV in an Unknown Environment. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 631. [CrossRef]
10. Liu, S.; Li, S.; Pang, L.; Hu, J.; Chen, H.; Zhang, X. Autonomous Exploration and Map Construction of a Mobile Robot Based on the TGHM Algorithm. *Sensors* **2020**, *20*, 490. [CrossRef] [PubMed]
11. Zhang, Z.; Shi, C.; Zhu, P.; Zeng, Z.; Zhang, H. Autonomous Exploration of Mobile Robots via Deep Reinforcement Learning Based on Spatiotemporal Information on Graph. *Appl. Sci.* **2021**, *11*, 8299. [CrossRef]
12. Prestes, E.; Idiart, M.; Engel, P.; Trevisan, M. Exploration technique using potential fields calculated from relaxation methods. In Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180), Maui, HI, USA, 29 October–3 November 2001; Volume 4, pp. 2012–2017. [CrossRef]

13. Maffei, R.; Jorge, V.A.M.; Prestes, E.; Kolberg, M. Integrated exploration using time-based potential rails. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 3694–3699. [CrossRef]
14. Shade, R.; Newman, P. Choosing where to go: Complete 3D exploration with stereo. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2806–2811. [CrossRef]
15. Jorge, V.A.M.; Maffei, R.; Franco, G.S.; Daltrozo, J.; Giambastiani, M.; Kolberg, M.; Prestes, E. Ouroboros: Using potential field in unexplored regions to close loops. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2125–2131. [CrossRef]
16. Silveira, R.; Prestes, E.; Nedel, L. Fast path planning using multi-resolution boundary value problems. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4710–4715. [CrossRef]
17. Wurm, K.M.; Stachniss, C.; Burgard, W. Coordinated multi-robot exploration using a segmentation of the environment. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 1160–1165. [CrossRef]
18. Cipolleschi, R.; Giusto, M.; Li, A.Q.; Amigoni, F. Semantically-informed coordinated multirobot exploration of relevant areas in search and rescue settings. In Proceedings of the 2013 European Conference on Mobile Robots, Barcelona, Catalonia, Spain, 25–27 September 2013; pp. 216–221. [CrossRef]
19. Basilico, N.; Amigoni, F. Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. *Auton. Robot.* **2011**, *31*, 401–417. [CrossRef]
20. Rekleitis, I. Multi-robot simultaneous localization and uncertainty reduction on maps (MR-SLURM). In Proceedings of the 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, 12–14 December 2013; pp. 1216–1221. [CrossRef]
21. Juliá, M.; Gil, A.; Reinoso, O. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Auton. Robot.* **2012**, *33*, 427–444. [CrossRef]
22. Mukhija, P.; Krishna, K.M.; Krishna, V. A two phase recursive tree propagation based multi-robotic exploration framework with fixed base station constraint. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4806–4811. [CrossRef]
23. Rooker, M.N.; Birk, A. Multi-robot exploration under the constraints of wireless networking. *Control. Eng. Pract.* **2007**, *15*, 435–445. [CrossRef]
24. Reich, J.; Misra, V.; Rubenstein, D.; Zussman, G. Connectivity Maintenance in Mobile Wireless Networks via Constrained Mobility. *IEEE J. Sel. Areas Commun.* **2012**, *30*, 935–950. [CrossRef]
25. Hollinger, G.A.; Singh, S. Multirobot Coordination With Periodic Connectivity: Theory and Experiments. *IEEE Trans. Robot.* **2012**, *28*, 967–973. [CrossRef]
26. Visser, A.; Slamet, B.A. Including communication success in the estimation of information gain for multi-robot exploration. In Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, Berlin, Germany, 31 March–4 April 2008; pp. 680–687. [CrossRef]
27. de Hoog, J.; Cameron, S.; Visser, A. Role-Based Autonomous Multi-robot Exploration. In Proceedings of the 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, Athens, Greece, 15–20 November 2009; pp. 482–487. [CrossRef]
28. Arkin, R.; Diaz, J. Line-of-sight constrained exploration for reactive multiagent robotic teams. In Proceedings of the 7th International Workshop on Advanced Motion Control. Proceedings (Cat. No.02TH8623), Maribor, Slovenia, 3–5 July 2002; pp. 455–461. [CrossRef]
29. Jensen, E.A.; Lowmanstone, L.; Gini, M.; Kolling, A.; Berman, S.; Frazzoli, E.; Martinoli, A.; Matsuno, F.; Gauci, M. Communication-Restricted Exploration for Search Teams. In *Distributed Autonomous Robotic Systems: The 13th International Symposium, Tokyo, Japan, 12–14 November 2018*; Springer: Cham, Switzerland, 2018; pp. 17–30. [CrossRef]
30. Lin, Z.; Zhang, S.; Yan, G. An incremental deployment algorithm for wireless sensor networks using one or multiple autonomous agents. *Ad Hoc Netw.* **2013**, *11*, 355–367. [CrossRef]
31. Stump, E.; Michael, N.; Kumar, V.; Isler, V. Visibility-based deployment of robot formations for communication maintenance. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Wellington, New Zealand, 6–8 December 2011; pp. 4498–4505. [CrossRef]
32. Pei, Y.; Mutka, M.W.; Xi, N. Connectivity and bandwidth-aware real-time exploration in mobile robot networks. *Wirel. Commun. Mob. Comput.* **2013**, *13*, 847–863. [CrossRef]
33. Grontas, P.D.; Vlantis, P.; Bechlioulis, C.P.; Kyriakopoulos, K.J. Computationally Efficient Harmonic-Based Reactive Exploration. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2280–2285. [CrossRef]
34. Chunyu, J.; Qu, Z.; Pollak, E.; Falash, M. Reactive target-tracking control with obstacle avoidance of unicycle-type mobile robots in a dynamic environment. In Proceedings of the American Control Conference, ACC 2010, Baltimore, MD, USA, 30 June–2 July 2010. [CrossRef]
35. Loizou, S.G. Closed form Navigation Functions based on harmonic potentials. In Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 12–15 December 2011; pp. 6361–6366. [CrossRef]

36. Karras, G.C.; Bechlioulis, C.P.; Fourlas, G.K.; Kyriakopoulos, K.J. Target Tracking with Multi-rotor Aerial Vehicles based on a Robust Visual Servo Controller with Prescribed Performance. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 480–487. [CrossRef]
37. Mahony, R.; Kumar, V.; Corke, P. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. *IEEE Robot. Autom. Mag.* **2012**, *19*, 20–32. [CrossRef]
38. Kohlbrecher, S.; Meyer, J.; von Stryk, O.; Klingauf, U. A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In Proceedings of the IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Kyoto, Japan, 1–5 November 2011.
39. Solomon, C.; Breckon, T. Front Matter. In *Fundamentals of Digital Image Processing*; John Wiley and Sons, Ltd.: Hoboken, NJ, USA, 2010. [CrossRef]
40. Gonzalez-Arjona, D.; Sanchez, A.; López-Colino, F.; De Castro, A.; Garrido, J. Simplified Occupancy Grid Indoor Mapping Optimized for Low-Cost Robots. *ISPRS Int. J. Geo-Inf.* **2013**, *2*, 959–977. [CrossRef]
41. Saad, Y.; Schultz, M.H. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* **1986**, *7*, 856–869. [CrossRef]
42. Yu, Y.; Yang, S.; Wang, M.; Li, C.; Li, Z. High performance full attitude control of a quadrotor on SO(3). In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1698–1703. [CrossRef]
43. Shamshirgaran, A.; Ebeigbe, D.; Simon, D. Position and Attitude Control of Underactuated Drones Using the Adaptive Function Approximation Technique. In Proceedings of the Dynamic Systems and Control Conference 2020, Online, 5–7 October 2020; American Society of Mechanical Engineers: New York, NY, USA, 2020; Volume 1, p. V001T01A004. [CrossRef]
44. Fenwick, J.; Newman, P.; Leonard, J. Cooperative concurrent mapping and localization. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), Washington, DC, USA, 11–15 May 2002; Volume 2, pp. 1810–1817. [CrossRef]
45. Howard, A. Multi-robot Simultaneous Localization and Mapping using Particle Filters. *Int. J. Robot. Res.* **2006**, *25*, 1243–1256. [CrossRef]
46. Thrun, S. A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots. *Int. J. Robot. Res.* **2001**, *20*, 335–363. [CrossRef]
47. Hörner, J. Map-Merging for Multi-Robot System. Bachelor's Thesis, Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic, 2016.
48. Citroni, R.; Di Paolo, F.; Livreri, P. A Novel Energy Harvester for Powering Small UAVs: Performance Analysis, Model Validation and Flight Results. *Sensors* **2019**, *19*, 1771. [CrossRef] [PubMed]
49. Liu, X.; Gong, D. A comparative study of A-star algorithms for search and rescue in perfect maze. In Proceedings of the International Conference on Electric Information and Control Engineering, Yichang, China, 16–18 September 2011; pp. 24–27. [CrossRef]
50. Rousseas, P.; Bechlioulis, C.P.; Kyriakopoulos, K.J. Trajectory Planning in Unknown 2D Workspaces: A Smooth, Reactive, Harmonics-Based Approach. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1992–1999. [CrossRef]