A Sampling-Based Approach to UAV Manipulator Path Planning

Zamoum Housseyn[1][a], Guiatni Mohamed[2][b], Bouzid Yasser[2][c], Alouane Mohamed Amine[2] and Khelal Atmane[1]

[1]*Guidance and Navigation Laboratory, Ecole Militaire Polytechnique, Algiers, Algeria*

[2]*Control of Complex Systems and Simulators Laboratory, Ecole Militaire Polytechnique, Algiers, Algeria*

Keywords: Path Planning, Sampling-Based Methods, Random Geometric Model, UAMs, UAV Manipulator.

Abstract: This paper presents a new approach to path planning for unmanned aerial manipulator systems (UAMs) using Sampling-Based Methods and Random Geometric Models (RGM) to efficiently search the configuration space for feasible, collision-free paths. The RGM generates random points in the UAM's workspace to guide sampling-based algorithms in constructing graphs that link the aerial manipulator's initial and final positions. These graphs are then explored using the RRT* algorithm to find an optimal collision-free path. The effectiveness of this approach is demonstrated through different scenarios, showing that it outperforms existing path planning techniques in terms of efficiency, computing time, and robustness. The proposed framework is adaptable to various application scenarios and environments, making it a valuable tool for applications such as search and rescue missions, surveillance, and inspection tasks.

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) equipped with manipulator's arms have become increasingly popular in recent years, offering a versatile and efficient solution for various applications, including pick up, place and transportation of objects, smart agriculture, object manipulation, inspection, and construction, in environments that are difficult or unsafe for humans to access (Ruggiero et al., 2018).

Generating efficient, collision-free trajectories for UAV manipulators is one of the main challenges in their operation, as it is critical for performing complex tasks accurately and quickly. However, path planning for UAV manipulators is a difficult problem due to several factors. The physical design of the aerial manipulation system may limit its range of motion due to size constraints or design choices made during system development. Additionally, the high degree of freedom and complex manipulator dynamics further complicate the path-planning process. These factors make it challenging to develop trajectory-planning algorithms that can handle the complex nature of UAV manipulators while generating efficient, collision-free paths.

Recently, there has been a growing interest in the use of sampling-based algorithms to solve the problem of trajectory planning for aerial manipulators. In their work, M. Brunner et al. (Brunner et al., 2022) proposed a sampling-based approach for fully-actuated MAVs that can exert forces and torques in six degrees of freedom (6 DoF). This makes them well-suited for aerial manipulation tasks that require precise control. The sampling-based approach utilized in their work does not rely on an analytical model of the interaction dynamics. This allows the approach to handle multiple and recurring contacts between the aerial manipulator and the environment. In (Yavari et al., 2022), M. Yavari proposes an integrated planning strategy for object retrieval using a sampling-based approach called Lazy-Steering-RRT*. This approach plans the motion of an Unmanned Aerial Manipulator (UAM) from its starting point to a pregrasp state while minimizing the motion of the arm. Limited-range sensors onboard facilitate on-the-fly planning using Machine-Learning-based. The authors in (Kim et al., 2019) propose a path planning approach for an aerial pick-and-place task, where an aerial manipulator is required to pick up or place an object at specified waypoints with partial state variable constraints. The proposed framework is based on the informed RRT* algorithm in a bidirectional man-

[a] https://orcid.org/0000-0002-3971-7426
[b] https://orcid.org/0000-0002-5899-6862
[c] https://orcid.org/0000-0002-8400-9912

Housseyn, Z., Mohamed, G., Yasser, B., Amine, A. and Atmane, K.
A Sampling-Based Approach to UAV Manipulator Path Planning.
DOI: 10.5220/0012761700003758
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 14th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2024)*, pages 80-90
ISBN: 978-989-758-708-5; ISSN: 2184-2841
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

ner and incorporates an extra merging process to integrate the trees from the start and goal points. In (Caballero et al., 2018), an airborne robotic system with two arms and a lengthy bar extension is designed for long-range manipulation in crowded settings. The authors provide detailed information on the trajectory planning performance of this system using a planner based on the Rapidly-exploring Random Tree (RRT*) algorithms. In (Lee et al., 2015), the authors present the planning of multiple aerial manipulators for cooperative transport. The desired path for each aerial manipulator is obtained by using RRT* to transport an object to the desired position while taking into consideration the constraints of the end effector's capture point.

Ying Gaoyang et al. introduced an enhanced version of the RRT algorithm called IRRT, which improves the selection strategy of the root node and incorporates trajectory distance constraints to enhance the quality of the planned trajectory (YIN et al., 2017). However, this algorithm still faces challenges related to high path cost and slow convergence speed. To address these issues, Jiaming Fan et al. incorporated a goal biasing strategy and a dichotomy-based approach to create a more goal-oriented sampling method in the RRT* algorithm, resulting in the PF-RRT* algorithm. This modification improves planning efficiency and search speed (Fan et al., 2022). Another approach proposed by Huang J et al. is the Bi-directional-Rapidly Exploring Random Tree (Bi-RRT*) algorithm, which avoids invalid expansion, reduces storage requirements, and enhances convergence speed (Huang and Sun, 2020). However, it may encounter difficulties in extremely complex environments. In conclusion, while the RRT algorithm is a widely used path planning algorithm, adjustments and optimizations are still necessary to improve its efficiency and effectiveness in various scene environments with different complexities.

Despite their effectiveness, conventional sampling-based techniques for aerial manipulator systems have limitations. These techniques do not incorporate the geometric model of the robot because they are designed to operate in a probabilistic framework.

This paper proposes a new path-planning method for UAMs that utilizes sampling-based algorithms and the direct geometric model (DGM) and inverse geometric model (IGM) equations to describe the coupled motions between the UAV and its manipulator's arm. The proposed approach combines the strengths of sampling-based methods and the RGM, which can be used together with DGM and IGM to generate efficiently achievable paths that can navigate complex and cluttered environments. Furthermore, this algorithm can efficiently explore high-dimensional spaces and find optimal solutions in terms of computation time and robustness compared to conventional methods in the literature.

The rest of the document is organized as follows: Section 2 presents the mathematical formulation of the path planning problem and the direct and inverse geometric modelling of UAMs. In Section 3, the proposed path planning strategy and the details of the algorithm developed in this work are described. Section 4 presents the different results obtained from implementing the algorithm for several scenarios. Finally, Section 5 concludes with the results and recommendations for future research.

2 MATHEMATICAL FORMULATIONS OF THE PROBLEM

Path planning for a UAV with a manipulator's arm involves finding a collision-free path that the system can follow from an initial position to a desired goal position. The mathematical formulation of this problem can vary depending on the specific scenario and objectives, but here are some of the common mathematical formulations used in path planning:

- Let $X \subseteq \mathbb{R}^3$ is the state space of the planning problem.
- The states that collide with the obstacles are called $X_{\text{obs}} \subset X$, and $X_{\text{dead}} \subset X$ is the dead space or "unreachable space", the states in which an aerial manipulator cannot manipulate an object where it is unable to reach it due to limitations in its physical structure or the constraints of the task.
- Let $X_{\text{free}} = X \backslash X_{\text{obs}} \cap X \backslash X_{\text{dead}}$ is the set of admissible states that results for the aerial robot, the initial state is X_{start}, and the desired finale state is X_{goal}.

Definition 1. A feasible path is defined for an aerial manipulator system, if it is a collision-free path and if there is a sequence of states $\sigma = [\sigma_{\mathcal{B}}, \sigma_e]$ defined as follows:

$$\begin{aligned} \sigma : [0,1] \times [0,1] &\longrightarrow X_{\text{free}} \times X_{\text{free}} \\ \lambda &\longrightarrow \sigma(\lambda) = [\sigma_{\mathcal{B}}, \sigma_e] \end{aligned} \tag{1}$$

with $\sigma(0,0) = [X_{\mathcal{B},\text{start}}, X_{e,\text{start}}]$ and $\sigma(1,1) = [X_{\text{final}}, X_{\text{goal}}]$.

The search for the optimal path σ_e^* and $\sigma_{\mathcal{B}}^*$, which minimizes a specified cost function, $c : \Gamma \longrightarrow \mathbb{R}^+$, which connects X_{start} to X_{goal} through the free space

X_{free} is the formal definition of the optimal planning problem and is expressed as follows:

$$\sigma^*_{\mathcal{B}} = \arg\min_{\sigma_{\mathcal{B}}\in\Gamma}\{c(\sigma_{\mathcal{B}}) \mid \forall s \in [0,1], \sigma_{\mathcal{B}}(t) \in X_{\text{free}}\} \quad (2)$$

Sampling-based methods such as RRT and RRT* are helpful for generating feasible paths that satisfy constraints such as obstacle avoidance. However, sampling-based methods cannot consider the UAV's positioning model or unreachable positions, as shown in Fig.1.

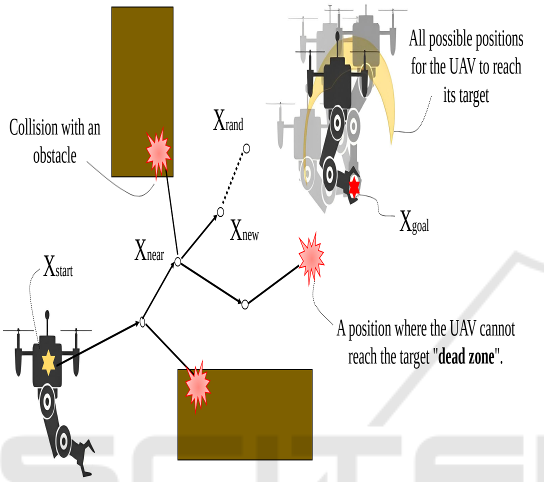

All possible positions for the UAV to reach its target
Collision with an obstacle
X_{rand}
X_{goal}
X_{new}
X_{near}
X_{start}
A position where the UAV cannot reach the target "dead zone".

Figure 1: The challenges of path planning through the sampling approaches of an aerial manipulator.

We consider a UAM which is a combined system of a UAV and an n-DoF robotic arm. The earth fixed frame $\{I\} = (O_I, X_I, Y_I, Z_I)$, the body fixe frame $\{\mathcal{B}\} = (O_{\mathcal{B}}, X_{\mathcal{B}}, Y_{\mathcal{B}}, Z_{\mathcal{B}})$ attached to the centre of mass of UAV, and the $\{\mathcal{B}_j\} = (O_j, X_j, Y_j, Z_j)$ reference frame. attached to the link j of the arm are defined in Fig. 2.

The position of $\{\mathcal{B}\}$ attached to the centre of UAV with respect to $\{I\}$ is given by the vector $X_{\mathcal{B}} = {}^{I}\mathcal{P}_{\mathcal{B}} = [xyz]^T$ and $X_e = {}^{I}\mathcal{P}_e = [x_e y_e z_e]^T$ is the position of the end effector relative to the reference frame $\{\mathcal{B}\}$. The system state can be decomposed into two vectors:

The first vector $\chi = \begin{bmatrix} X_e^T & \Phi_e^T \end{bmatrix}^T$, where $\Phi_e = [\phi_e \ \theta_e \ \psi_e]^T$ expressed as the generalized coordinates of the terminal organ attached to the frame $\{\mathcal{B}_e\}$.

The second vector represented by the collection of state variables of the UAV and the arm joint as $\xi = \begin{bmatrix} X_{\mathcal{B}}^T & \Phi_{\mathcal{B}}^T & Q_a^T \end{bmatrix}^T$, with $\Phi_{\mathcal{B}} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ gives the orientation of the UAV with respect to the fixed reference frame $\{I\}$, $Q_a = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}^T$ represent the joint vector of the n-DoF manipulator's arm expressed in the reference frame $\{\mathcal{B}\}$.

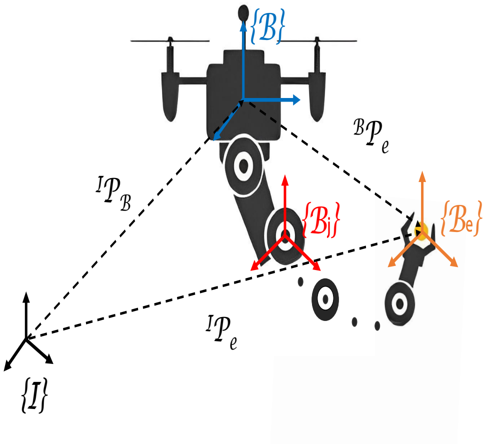


Figure 2: Configuration space of an unmanned aerial manipulator (i.e. a UAV with a robot arm with n-DoF) in our proposed configurations.

2.1 Direct Geometrical Models

The direct geometrical problem consists in determining the operational coordinates ${}^{I}P_e$ of the end effector, according to the movements of the UAV as well as the movement of the manipulator's joints.

Definition 2. Consider $\mathcal{D}_{\mathcal{G}}$ as a mathematical function describing the transition from the operational coordinate space $\mathcal{N} \subseteq \mathbb{R}^6$ to the joint space $\mathcal{M} \subseteq \mathbb{R}^{6+n}$ as follows:

$$\begin{aligned} \mathcal{D}_{\mathcal{G}} : \mathcal{N} &\longrightarrow \mathcal{M} \\ \xi &\longrightarrow \chi = \mathcal{D}_{\mathcal{G}}(\xi) \end{aligned} \quad (3)$$

The DGM problem is solved geometrically using the following system of equations:

$$\begin{cases} {}^{I}P_e = {}^{I}P_{\mathcal{B}} + {}^{I}\mathcal{R}_{\mathcal{B}}(\Phi_{\mathcal{B}}) * {}^{\mathcal{B}}P_e(Q_a) \\ {}^{I}\mathcal{R}_e(\Phi_e) = {}^{I}\mathcal{R}_{\mathcal{B}}(\Phi_{\mathcal{B}}) * {}^{\mathcal{B}}\mathcal{R}_e(Q_a) = \mathcal{R}(\Phi_{\mathcal{B}}, Q_a) \end{cases} \quad (4)$$

where represent where ${}^{b}\mathcal{R}_a$ is the rotation matrix from frame $\{a\}$ to frame $\{b\}$.

Based on the system of equations 4 the function $\mathcal{D}_{\mathcal{G}}$ can be given as follows:

$$\mathcal{D}_{\mathcal{G}} = \begin{bmatrix} \mathcal{D}_{\mathcal{G}}^{+} \\ \mathcal{D}_{\mathcal{G}}^{-} \end{bmatrix} = \begin{bmatrix} X_e \\ \Phi_e \end{bmatrix} = \begin{bmatrix} X_{\mathcal{B}} + \mathcal{G}(\Phi_{\mathcal{B}}, Q_a) \\ \mathcal{H}(\Phi_{\mathcal{B}}, Q_a) \end{bmatrix}. \quad (5)$$

with the function $\mathcal{G}(\Phi_{\mathcal{B}}, Q_a) = {}^{I}\mathcal{R}_{\mathcal{B}}(\Phi_{\mathcal{B}}) * {}^{\mathcal{B}}P_e(Q_a)$ and the function $\mathcal{H}(\Phi_{\mathcal{B}}, Q_a)$ is defined as:

$$\mathcal{H}(\Phi_{\mathcal{B}}, Q_a) = \begin{cases} atan2\left(\mathcal{R}(\Phi_{\mathcal{B}}, Q_a)_{32}, \mathcal{R}(\Phi_{\mathcal{B}}, Q_a)_{33}\right) \\ atan2\left(-\mathcal{R}(\Phi_{\mathcal{B}}, Q_a)_{31}, \sqrt{\mathcal{R}_{32}^2 + \mathcal{R}_{33}^2}\right) \\ atan2\left(\mathcal{R}(\Phi_{\mathcal{B}}, Q_a)_{21}, \mathcal{R}(\Phi_{\mathcal{B}}, Q_a)_{11}\right) \end{cases} \quad (6)$$

where $\mathcal{R}(\Phi_{\mathcal{B}}, Q_a)_{ij}$ represents the elements of the matrix $\mathcal{R}(\Phi_{\mathcal{B}}, Q_a)$.

To go further, the equation (5) shows all the suitable positions of the drone to reach its target with its end effector for different values of $\Phi_{\mathcal{B}}$ and Q_a. This result is then used in the trajectory planning to guide the UAV to the appropriate region all based on the $\mathcal{D}_{\mathcal{G}}^{+}$-function.

2.2 Inverse Geometrical Models

The inverse geometric approach is fundamental to the control of the UAM as it allows the calculation of the UAV motions and the manipulator arm joint variables required to move the end effector to a desired position.

Definition 3. Consider $I_{\mathcal{G}}$ as a mathematical function describing the transition from the joint space $\mathcal{M}$ to the operational coordinate space $\mathcal{N}$ as follows:

$$\begin{aligned} I_{\mathcal{G}} : \mathcal{M} &\longrightarrow \mathcal{N} \\ \chi &\longrightarrow \xi = I_{\mathcal{G}}(\chi) \end{aligned} \tag{7}$$

The inverse geometry issue entails calculating the X motions of the UAV as well as the joints of the manipulator Q_a. However, let's consider equation (5). We have a system of non-linear equations of several variables from $\mathbb{R}^6 \to \mathbb{R}^{6+n}$. The inverse analytical resolution of this system is quite complex due to several factors: the problem to be solved is a system of generally nonlinear equations, several solutions can be found, and no solutions can be found after an analytical mathematical calculation.

The General Paul Inversion method is a mathematical approach used to solve the inverse geometric problem for robotic manipulators. This approach used homogeneous transformation matrices ${}^{j}T_i$ to achieve the desired position and orientation of the end effector given by the matrix $\mathcal{U}({}^{I}\mathcal{R}_e(\Phi_{e_d}), {}^{I}P_{e_d})$ as follows:

$$\mathcal{U}({}^{I}\mathcal{R}_e(\Phi_{e_d}), {}^{I}P_{e_d}) = \begin{pmatrix} {}^{I}\mathcal{R}_e(\Phi_{e_d}) & {}^{I}P_{e_d} \\ 0_{1\times 3} & 1 \end{pmatrix} \tag{8}$$

The passage from the frame of reference attached to the end organ $\{\mathcal{B}_e\}$ to the inertial frame $\{I\}$, is provided by the matrix:

$$\mathcal{U} = {}^{I}T_e = {}^{I}T_{\mathcal{B}}\,{}^{\mathcal{B}}T_1\,{}^{1}T_2\,{}^{2}T_3 \dots {}^{n}T_e \tag{9}$$

we use this equation (9) to compute the manipulator arm state variables Q_a using the following recursive method:

$$\left.\begin{array}{l} \mathcal{U} = {}^{I}T_{\mathcal{B}}\,{}^{\mathcal{B}}T_1\,{}^{1}T_2\,{}^{2}T_3 \dots {}^{n}T_e \\ {}^{\mathcal{B}}T_I\,\mathcal{U} = {}^{\mathcal{B}}T_1\,{}^{1}T_2\,{}^{2}T_3 \dots {}^{n}T_e \\ {}^{1}T_{\mathcal{B}}\,{}^{\mathcal{B}}T_I\,\mathcal{U} = {}^{1}T_2\,{}^{2}T_3 \dots {}^{n}T_e \\ {}^{2}T_1\,{}^{1}T_{\mathcal{B}}\,{}^{\mathcal{B}}T_I\,\mathcal{U} = {}^{2}T_3 \dots {}^{n}T_e \\ \vdots \\ {}^{n}T_{n-1} \dots {}^{2}T_1\,{}^{1}T_{\mathcal{B}}\,{}^{\mathcal{B}}T_I\,\mathcal{U} = {}^{n}T_e \end{array}\right\} \longrightarrow Q_a \tag{10}$$

With the assumption of hovering manipulation $(\phi = \theta = 0)$ we can easily deduce the vector $Q_a = \mathcal{L}(X_e, \Phi_e)$ replacing Φ_{e_d} with Φ_e, and X_e with X_{e_d}. Then to determine the position of the UAV presented by the vector $X_{\mathcal{B}}$ we use the function $\mathcal{D}_{\mathcal{G}}^{+}$ developed in equation (5), hence the position expression of the UAV is $X_{\mathcal{B}} = X_e - \mathcal{G}(\Phi_{\mathcal{B}}, Q_a)$.

Finally, the function $I_{\mathcal{G}}$ described the IGM can be given as follows:

$$I_{\mathcal{G}} = \begin{bmatrix} I_{\mathcal{G}}^{+} \\ \Phi_{\mathcal{B}} \\ I_{\mathcal{G}}^{-} \end{bmatrix} = \begin{bmatrix} X_{\mathcal{B}} \\ \Phi_{\mathcal{B}}^{0} \\ Q_a \end{bmatrix} = \begin{bmatrix} X_e - \mathcal{G}(\Phi_{\mathcal{B}}^{0}, Q_a) \\ \Phi_{\mathcal{B}}(0, 0, \psi_d) \\ \mathcal{L}(X_e, \Phi_e) \end{bmatrix} \tag{11}$$

One advantage of the Inversion method is that it provides a mathematical solution to the inverse geometric problem for UAV manipulators. However, this method may disregard constraints such as obstacle avoidance.

3 PATH PLANNING STRATEGY

Path planning is a critical component of the control system used in aerial manipulation. It involves the creation of a path through obstacle avoidance, which an autonomous vehicle should follow to reach a particular destination. Its benefits include increased safety, efficiency and autonomy. These advantages help to reduce operational costs and ensure the efficient use of UAM.

Sampling-based planners' algorithms have demonstrated high potential in finding fast solutions for high-dimensional robots. Furthermore, some of these methods bring the possibility of generating motion plans that optimize certain cost functions and use heuristics guide, as for the Informed RRT* (Gammell et al., 2014). The algorithm is an extension of the RRT algorithm, which uses heuristics to guide the exploration of the RRT tree to the goal location using information about the distance to the goal and the quality of the paths that have been explored so far. These techniques allow Informed RRT* to find an optimal path to the goal in a more efficient manner than the original RRT algorithm.

In this work, trajectory planning for a manipulator UAV using sampling and random geometric model (RGM) based methods is a technique used to generate feasible trajectories for an unmanned aerial vehicle (UAV) with a manipulator's arm.

This technique is divided into two steps, in the first step the sampling is based on RRT* to ensure optimal search in the working space where the target object is located while avoiding obstacles. In the second stage,

```
$\mathcal{V} \leftarrow \{X_{\mathcal{B},\text{star}}, X_{e,\text{star}}\}; E \leftarrow \{\emptyset\}$;
$\mathcal{T} \leftarrow \{\mathcal{V}, E\}$;
for $k = 1$ to $N$ do
    $X_{\mathcal{B},\text{RGM}} \leftarrow \text{RGM}(X_{\text{goal}})$;
    $X_{\mathcal{B},\text{nearest}} \leftarrow \text{Nearest}(\mathcal{T}, X_{\text{rand}})$;
    $X_{\mathcal{B},\text{new}} \leftarrow \text{Steer}(X_{\mathcal{B},\text{nearest}}, X_{\mathcal{B},\text{rand}})$;
    if $CollisionFree(X_{\mathcal{B},nearest}, X_{\mathcal{B},new}, map)$ then
        $X_{\mathcal{B},\text{near}} \leftarrow \text{Near}(\mathcal{T}, X_{\mathcal{B},\text{new}}, r_{RRT^*})$;
        $\mathcal{T} \leftarrow \text{Add}(X_{\mathcal{B},\text{nearest}}, X_{\mathcal{B},\text{near}}, X_{\mathcal{B},\text{new}})$;
        $\mathcal{T} \leftarrow \text{Rewire}(X_{\mathcal{B},\text{near}}, X_{\mathcal{B},\text{new}})$;
    else
        $X_{\mathcal{B},\text{rand}} \leftarrow \text{Random(map)}$;
        $X_{\mathcal{B},\text{nearest}} \leftarrow \text{Nearest}(\mathcal{T}, X_{\text{rand}})$;
        $X_{\mathcal{B},\text{new}} \leftarrow \text{Steer}(X_{\mathcal{B},\text{nearest}}, X_{\mathcal{B},\text{rand}})$;
        if $CollisionFree(X_{\mathcal{B},nearest}, X_{\mathcal{B},new}, map)$ then
            $X_{\mathcal{B},\text{near}} \leftarrow \text{Near}(\mathcal{T}, X_{\mathcal{B},\text{new}}, r_{RRT^*})$;
            $\mathcal{T} \leftarrow \text{Add}(X_{\mathcal{B},\text{nearest}}, X_{\mathcal{B},\text{near}}, X_{\mathcal{B},\text{new}})$;
            $\mathcal{T} \leftarrow \text{Rewire}(X_{\mathcal{B},\text{near}}, X_{\mathcal{B},\text{new}})$;
        end
    end
end
$\mathcal{T}^*_{\mathcal{B}}, \mathcal{T}^*_e \leftarrow \text{DualPath}(\mathcal{T}, X_{\text{goal}}, \Phi_{e,\text{goal}})$;
return $\mathcal{T}^*_{\mathcal{B}}, \mathcal{T}^*_e$;
```

Algorithm 1: S-RGM*$(X_{\mathcal{B},\text{start}}, X_{e,\text{start}}, X_{\text{goal}}, map)$.

the sampling introduces the RGM which provides information on the system geometry to guide the sampling to converge quickly to the position where the UAM is able to position its body and manipulate it with its arm to reach the target objects while ensuring a safe transition between the DGM and the IGM using the functions $\mathcal{D}_g$ and I_g developed in section 2.

The proposed planning strategy is divided into two phases, as shown in Fig. 7. In the first phase, the search extends to the target object around the end point using RGM, as shown in Fig. 3 and 4. Once the target object is found, the algorithm proceeds to the final phase, which involves finding all possible positions of the UAM to pick up the object. All desired paths are clearly shown in Fig. 5 and 6. The Algorithm 1 presents the pseudocode for the suggested algorithm using the RGM.

In order to apply this general structure to the UAV manipulator system, some of the intermediate functionalities have been adapted to the studied problem. These particular developments will be discussed below.

3.1 The RGM Function

This function generates a random sample in the configuration space around the target object, using the IGM and a random value of $\Phi^0_{\mathcal{B}}, \mathcal{Q}_a$, all based on equations (12).

$$X_{\mathcal{B},\text{RGM}} = I^+_{\mathcal{G}}(X_{\text{goal}}) = X_{\text{goal}} - \mathcal{G}(\Phi^0_{\mathcal{B},\text{rand}}, \mathcal{Q}_{a,\text{rand}}) \tag{12}$$

As shown in equation (11), the RGM uses random values of the articulation variables $\mathcal{Q}_a$ and the attitude of the UAM $\Phi^0_{\mathcal{B}}$ to obtain all the possible positions of the drone so that it can be manipulated with its arm correctly (Fig.4 show the cloud of points generated by the RGM for two types of UAM, $\mathcal{H}$-RRR in (Zamoum et al., 2023) and $\mathcal{Q}$-PRR in (Bouzgou, 2021)).

The RGM in Algorithm 2 is a compact representation of the workspace of a UAM that can be used for efficient sampling for the manipulator arm that is likely to lead to a successful trajectory to the target.

```
$\Phi^0_{\mathcal{B},\text{rand}} \leftarrow \text{rand}(\Phi^0_{\text{Min}}, \Phi^0_{\text{Max}})$;
$\mathcal{Q}_{a,\text{rand}} \leftarrow \text{rand}(\mathcal{Q}_{a,\text{Min}}, \mathcal{Q}_{a,\text{Max}})$;
$X_{\mathcal{B},\text{RGM}} \leftarrow I^+_{\mathcal{G}}(X_{\text{goal}}, \Phi^0_{\mathcal{B},\text{rand}}, \mathcal{Q}_{a,\text{rand}})$;
return $X_{\mathcal{B},rand}$;
```

Algorithm 2: RGM(X_{goal}).

3.2 Nearest Neighbor Function

This function finds the nearest node in the tree $\mathcal{T}$ to a given configuration $X_{\mathcal{B},\text{RGM}}$ or $X_{\mathcal{B},\text{rand}}$. It calculates the distance between each node in $\mathcal{T}$ and $X_{\mathcal{B},\text{rand}}$ and returns the node with the smallest distance $X_{\mathcal{B},\text{nearest}}$.

3.3 The Steer Function

This function generates a new configuration $X_{\mathcal{B},\text{new}}$ by steering from the nearest node $X_{\mathcal{B},\text{nearest}}$ to configuration $X_{\mathcal{B},\text{rand}}$ towards a randomly generated configuration rand. The new configuration $X_{\mathcal{B},\text{new}}$ is generated by taking a small step in the direction of $X_{\mathcal{B},\text{rand}}$ while ensuring that the aerial manipulator remains collision-free.

3.4 Chek Collision with Obstacle

This function checks whether the path from the nearest node $X_{\mathcal{B},\text{nearest}}$ to configuration $X_{\mathcal{B},\text{rand}}$ to the new configuration $X_{\mathcal{B},\text{new}}$ is collision-free. It does this by checking for collisions between the aerial manipulator and obstacles in the environment represented by the map.

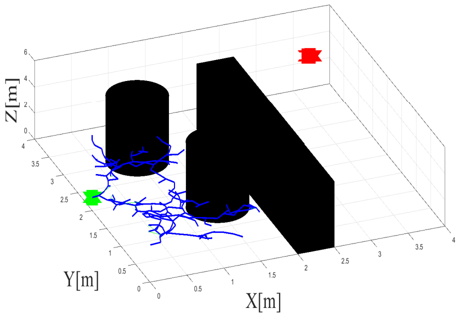


Figure 3: During each iteration in the first phase, the search extends to the target object around the goal point using RGM.

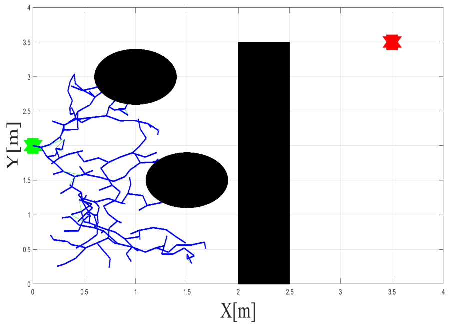


Figure 4: 2D representation of the first phase of the search for the target object.

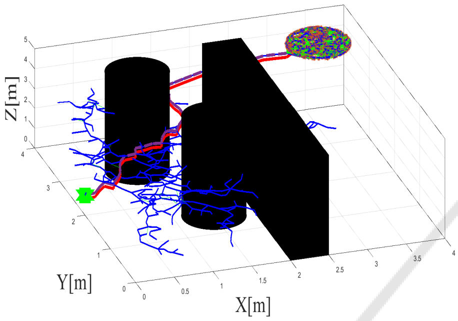


Figure 5: When the target object is found, the algorithm ends with the final phase to find all possible positions of the UAM to pick up the object.

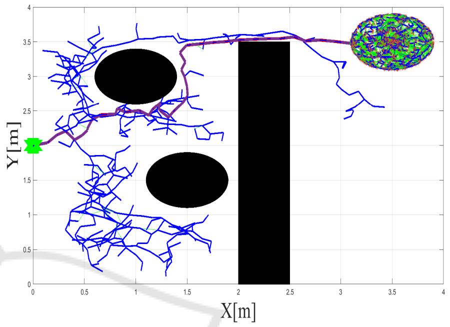


Figure 6: 2D representation of the final phase to pick up the object.

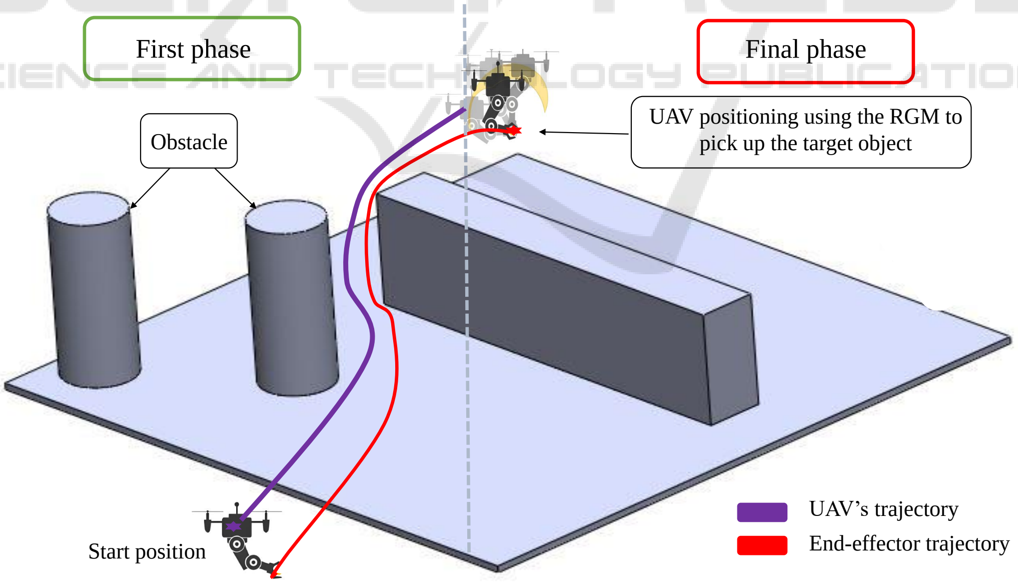


Figure 7: The path planning phases of the aerial manipulator that must pass to pick up the object. The start and end positions of the UAM are shown in a green and red star. Each purple and red line shows the trajectory of the aerial manipulator's body and effector position, respectively.

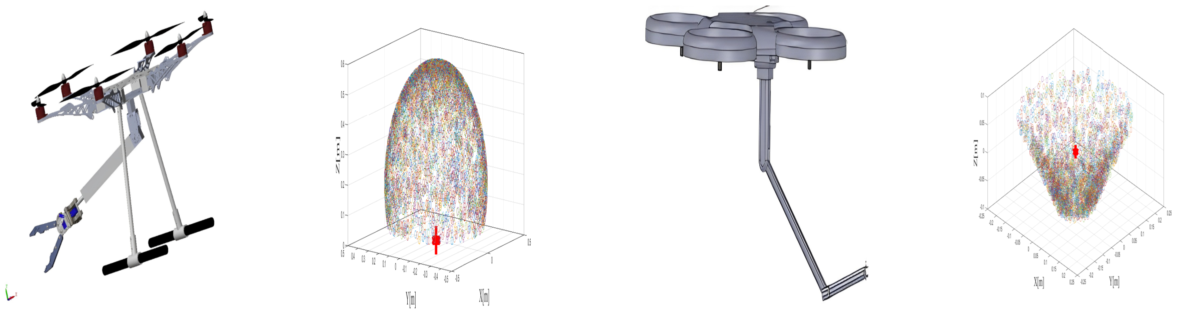


Figure 8: RGM cloud points for two types of UAM, $\mathcal{H}$-RRR and $\mathcal{Q}$-PRR.

3.5 The near Vertices Function

This function finds all nodes in the tree $\mathcal{T}$ within a certain radius r of a given configuration X. It calculates the distance between each node in $\mathcal{T}$ and X and returns the nodes with distances less than r.

3.6 Add Function

This function adds a new configuration $X_{\mathcal{B},\text{new}}$ to the tree $\mathcal{T}$ by connecting it to its nearest neighbour $X_{\mathcal{B},\text{nearest}}$ via a nearby node $X_{\mathcal{B},\text{near}}$. It creates a new edge in the tree between $X_{\mathcal{B},\text{near}}$ and $X_{\mathcal{B},\text{new}}$ through $X_{\mathcal{B},\text{near}}$.

```
$\mathcal{V} \leftarrow \mathcal{V} \cup \{X_{\mathcal{B},\text{new}}\}$;
$x_{min} \leftarrow X_{\mathcal{B},nearest}$;
$c_{min} \leftarrow Cost(X_{min}) + c(Line(x_{min}, X_{\mathcal{B},new})$;
for $\forall x_{near} \in X_{\mathcal{B},near}$ do
    $c_{new} \leftarrow$
      $Cost(x_{near}) + c(Line(x_{near}, X_{\mathcal{B},new}))$;
    if $c_{new} < c_{min}$ then
        if $CollisionFree(x_{near}, X_{\mathcal{B},new})$ then
            $x_{min} \leftarrow x_{near}$;
            $c_{min} \leftarrow c_{new}$;
        end
    end
end
$E \leftarrow E \cup (x_{min}; X_{\mathcal{B},new})$;
$\mathcal{T} \leftarrow \{\mathcal{V}, E\}$;
return $\mathcal{T}$;
```

Algorithm 3: Add($X_{\mathcal{B},\text{nearest}}, X_{\mathcal{B},\text{near}}, X_{\mathcal{B},\text{new}}$).

3.7 Rewire Function

This function rewires the tree $\mathcal{T}$ to ensure optimality. It checks if any nodes in the tree within a certain radius of $X_{\mathcal{B},\text{new}}$ can be reached with a shorter path via $X_{\mathcal{B},\text{new}}$. If so, it re-parents these nodes to $X_{\mathcal{B},\text{new}}$ and updates the cost of the path to each of these nodes.

```
for $\forall x_{near} \in X_{\mathcal{B},near}$ do
    $c_{near} \leftarrow Cost(x_{near})$;
    $c_{new} \leftarrow$
      $Cost(X_{\mathcal{B},new}) + c(Line(X_{\mathcal{B},new}, x_{near}))$;
    if $c_{new} < c_{near}$ then
        if $CollisionFree(X_{\mathcal{B},new}, x_{near})$ then
            $x_{parent} \leftarrow Parent(x_{near})$;
            $c_{min} \leftarrow c_{new}$;
        end
    end
    $E \leftarrow (E \setminus \{x_{parent}, x_{near}\}) \cup (X_{\mathcal{B},new}, x_{near})$;
end
$\mathcal{T} \leftarrow \{\mathcal{V}, E\}$;
return $\mathcal{T}$;
```

Algorithm 4: Rewire($X_{\mathcal{B},\text{near}}, X_{\mathcal{B},\text{new}}$).

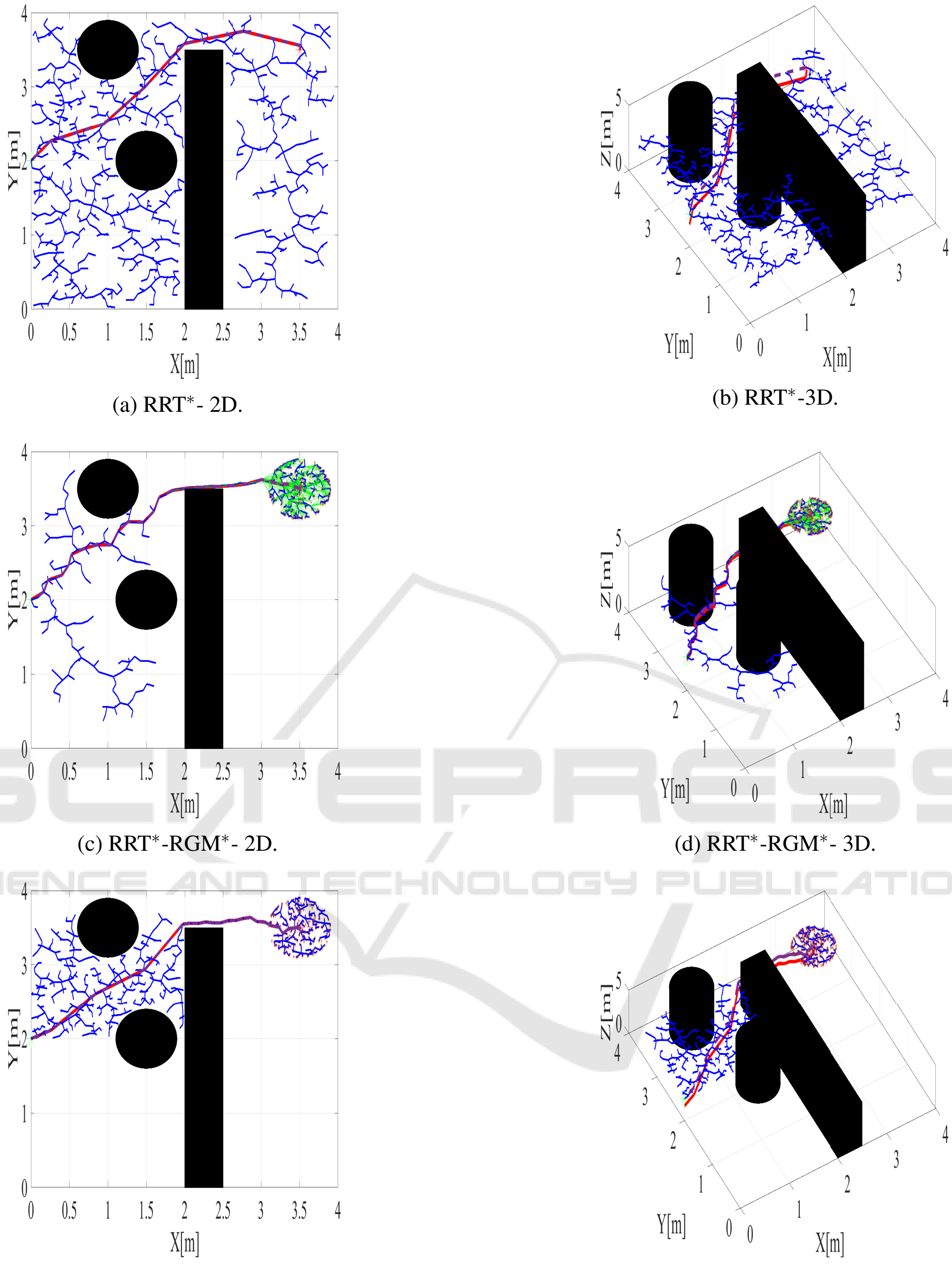
3.8 DualPath Function

This function finds the optimal paths from the start configuration $X_{\mathcal{B},\text{start}}$ to the final configuration $X_{\mathcal{B},\text{Final}}$, and from the configuration $X_{e,\text{start}}$ to the goal configuration X_{goal} with the desired end effector orientation $\Phi_{e,\text{goal}}$. It uses a bidirectional search algorithm to simultaneously search for a path from the start and end configurations and then merges the two paths $\mathcal{T}_{\mathcal{B}}^{*}$ and $\mathcal{T}_{e}^{*}$ to find the optimal solution.

```
$\mathcal{T}_{\mathcal{B}}^{*} \leftarrow \text{GetsPath}_{\mathcal{B}}(\mathcal{T})$;
$X_{\mathcal{B},\text{Final}} \leftarrow \text{Gets}(\mathcal{T}_{\mathcal{B}}^{*})$;
$Q_{a,\text{Final}} \leftarrow I_{\mathcal{G}}^{-}(X_{\text{goal}}, \Phi_{e,\text{goal}})$;
$X_{e,\text{Final}} \leftarrow \mathcal{D}_{\mathcal{G}}^{+}(X_{\mathcal{B},\text{Final}}, \Phi_{\mathcal{B}}^{0}, Q_{a,\text{Final}})$;
$\mathcal{V} \leftarrow \mathcal{V} \cup \{X_{\mathcal{B},\text{Final}}, X_{e,\text{Final}}\}$;
$\mathcal{T} \leftarrow \{\mathcal{V}, E\}$;
$\mathcal{T}_{e}^{*} \leftarrow \text{GetsPath}_{e}(\mathcal{T}, X_{e,\text{Final}})$;
return $\mathcal{T}_{\mathcal{B}}^{*}, \mathcal{T}_{e}^{*}$;
```

Algorithm 5: DualPath($\mathcal{T}, X_{\text{goal}}, \Phi_{e,\text{goal}}$).

In summary, this algorithm uses sampling-based methods to explore the configuration space of a UAV



(a) RRT*- 2D.

(b) RRT*-3D.

(c) RRT*-RGM*- 2D.

(d) RRT*-RGM*- 3D.

(e) S-RGM*- 2D.

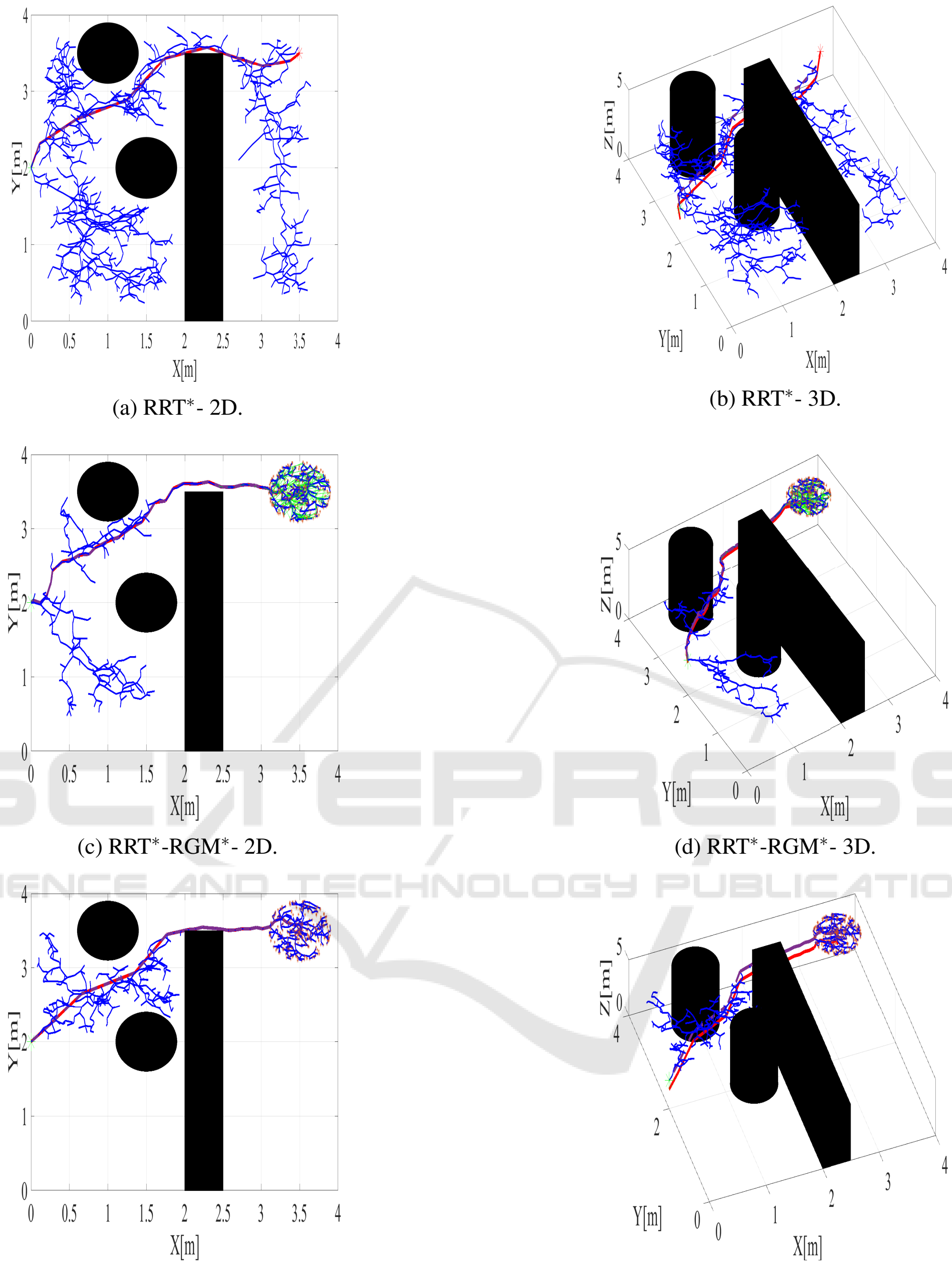
(f) S-RGM*- 3D.

Figure 9: The first case of path planning in the second scenario to pick up the object. Each purple and red line shows the trajectory of the aerial manipulator's body and effector position, respectively.

manipulator and plan a collision-free trajectory. The RGM is used to generate random samples, and the algorithm builds a tree of nodes to represent the explored configurations. The results of the S-RGM* algorithm are presented in the following section.

4 SIMULATION RESULTS AND DISCUSSION

To evaluate the efficacy of the proposed S-RGM* algorithm in path planning, we employed MATLAB as



(a) RRT*- 2D.

(b) RRT*- 3D.

(c) RRT*-RGM*- 2D.

(d) RRT*-RGM*- 3D.

(e) S-RGM*- 2D.

(f) S-RGM*- 3D.

Figure 10: The second case of path planning in the second scenario to pick up the object. Each purple and red line shows the trajectory of the aerial manipulator's body and effector position, respectively.

the simulation platform for this study. The simulation experiments were conducted in a three-dimensional environment. In two scenario, we examined a two-dimensional environment with obstacles and compared the path planning results of UAM using the RRT* and S-RGM* algorithms, analyzing search time and path length under identical conditions for a system ($\mathcal{H}$-RRR) under identical conditions.

The Two Scenarios

In this scenarios, the algorithm is deployed within an environment where obstacles impede the trajectory towards the desired object. To replicate this sce-

Table 1: Comparison performance table of three approaches RRT*, RRT*-RGM*, and S-RGM*in the second scenario.

Scenario	Approach	Time[S]	Iteration	Path length[UAV](m)
1	RRT*	82.2411	972	4.5432
	RRT*-RGM*	57.7834	547	4.2019
	S-RGM*	34.2245	475	4.0003
2	RRT*	144.5742	1312	5.7248
	RRT*-RGM*	66.4797	626	4.5432
	S-RGM*	38.3879	594	4.3307

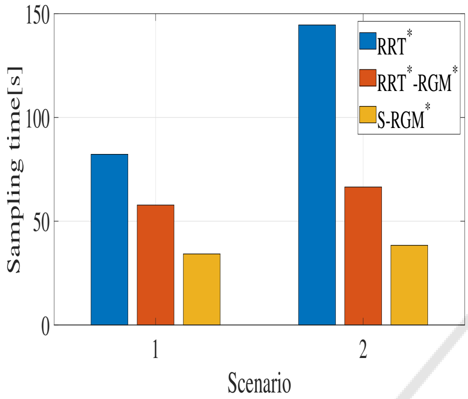


Figure 11: Comparison results of sampling time.

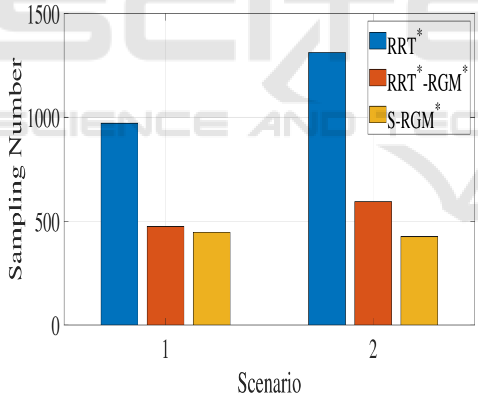


Figure 12: Comparison results of number of samples.

nario, three obstacles are introduced into the three-dimensional (3D) environment, as depicted in Fig.7. Within this scenario, two distinct cases are distinguished for the Unmanned Aerial Vehicle (UAM) to attain its target:

1. In the first case, the UAM executes a vertical maneuver (vertical takeoff) until it reaches the altitude of the desired object. Subsequently, the algorithm searches for the optimal path while maintaining a constant altitude. The algorithm undergoes 1500 iterations, and the resulting trees generated by the algorithm are illustrated in Figure 18.
2. In the second case, the algorithm directly pursues the optimal path, considering not only the horizontal coordinates (X and Y) but also the vertical dimension (Z). The algorithm operates for 2000 iterations, and the corresponding trees produced by the algorithm are depicted in Figure 19.

By differentiating between these cases, the algorithm aims to ascertain the most efficient path for the UAM, taking into account both the horizontal and vertical aspects of the environment.

Table 1 provides a comprehensive analysis of the performance of three distinct approaches: RRT*, RRT*-RGM*, and S-RGM*, within the context of the second scenario. The table showcases key metrics, including the sampling time, sampling number, and path lengths for both the UAV and the manipulator arm.

In the first case, the RRT* approach exhibited a sampling time of 82.24 seconds, using a total of 972 samples. It resulted in a UAV path length of 4.543 meters and an ARM path length of 4.94 meters. Conversely, the RRT*-RGM* approach achieved a slightly reduced sampling time of 57.78 seconds with 547 samples, producing a UAV path length of 4.20 meters and an ARM path length of 4.93 meters. Lastly, the S-RGM* approach showcased superior efficiency, utilizing a sampling time of 34.22 seconds with 475 samples, leading to a UAV path length of 4.00 meters and an ARM path length of 4.80 meters.

For scenario 2, the RRT* approach recorded a higher sampling time of 144.57 seconds, employing 1312 samples. This resulted in a UAV path length of 5.72 meters and an ARM path length of 6.12 meters. Conversely, the RRT*-RGM* approach demonstrated increased efficiency with a reduced sampling

time of 66.47 seconds and 626 samples. The resulting UAV path length was 4.54 meters, while the ARM path length measured 4.94 meters. Similarly, the S-RGM* approach showcased notable efficiency with a sampling time of 38.38 seconds, utilizing 594 samples, resulting in a UAV path length of 4.3307634876 meters and an ARM path length of 5.08 meters.

In summary, the S-RGM* approach exhibited superior performance across both scenarios, achieving shorter path lengths for both the UAV and the ARM. The RRT*-RGM* approach demonstrated improved efficiency in terms of sampling time (as showen in Fig.11) and sampling number (given in Fig.12). Meanwhile, the RRT* approach showcased higher path lengths for both the UAV and the ARM in scenario 2.

5 CONCLUSIONS

The article discusses the potential advantages of aerial robot manipulators, including their ability to manipulate objects in inaccessible, dangerous, or complex locations. The article proposes a solution for path planning using Sampling-Based Methods and the RGM. The results obtained through simulation have been satisfactory. The proposed solution provides a simple and effective way to plan trajectories for aerial robot manipulators, which could have significant practical applications in the future.

REFERENCES

Bouzgou, K. (2021). *Contribution à l'architecture, la modélisation et la commande d'un bras manipulateur aérien*. PhD thesis, Université Paris-Saclay.

Brunner, M., Rizzi, G., Studiger, M., Siegwart, R., and Tognon, M. (2022). A planning-and-control framework for aerial manipulation of articulated objects. *IEEE Robotics and Automation Letters*, 7(4):10689–10696.

Caballero, A., Suarez, A., Real, F., Vega, V. M., Bejar, M., Rodriguez-Castaño, A., and Ollero, A. (2018). First experimental results on motion planning for transportation in aerial long-reach manipulators with two arms. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8471–8477. IEEE.

Fan, J., Chen, X., Wang, Y., and Chen, X. (2022). Uav trajectory planning in cluttered environments based on pf-rrt* algorithm with goal-biased strategy. *Engineering Applications of Artificial Intelligence*, 114:105182.

Gammell, J. D., Srinivasa, S. S., and Barfoot, T. D. (2014). Informed rrt: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2997–3004. IEEE.

Huang, J. and Sun, W. (2020). A method of feasible trajectory planning for uav formation based on bidirectional fast search tree. *Optik*, 221:165213.

Kim, H., Seo, H., Kim, J., and Kim, H. J. (2019). Sampling-based motion planning for aerial pick-and-place. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7402–7408. IEEE.

Lee, H., Kim, H., and Kim, H. J. (2015). Path planning and control of multiple aerial manipulators for a cooperative transportation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2386–2391. IEEE.

Ruggiero, F., Lippiello, V., and Ollero, A. (2018). Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters*, 3(3):1957–1964.

Yavari, M., Gupta, K., and Mehrandezh, M. (2022). Interleaved predictive control and planning for an unmanned aerial manipulator with on-the-fly rapid replanning in unknown environments. *IEEE Transactions on Automation Science and Engineering*, pages 1–16.

YIN, G.-y., ZHOU, S.-l., and WU, Q.-p. (2017). An improved rrt algorithm for uav path planning. *ACTA ELECTONICA SINICA*, 45(7):1764.

Zamoum, H., Bouzid, Y., and Guiatni, M. (2023). Robust control and recursive modelling approach for hexarotor manipulator. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, page 09596518231156814.