Unmanned Systems, Vol. 14, No. 1 (2026) 119–141

DOI: 10.1142/S2301385025500852

Unmanned Systems
https://www.worldscientific.com/worldscinet/us

Path Planning Strategy for Unmanned Aerial Manipulators in Pick-and-Place Applications

Zamoum Housseyn*

Guidance and Navigation Laboratory,
Ecole Militaire Polytechnique,
Algiers, Algeria

Bouzid Yasser, Guiatni Mohamed

Control of Complex Systems and Simulators Laboratory,
Ecole Militaire Polytechnique, Algiers, Algeria

This paper introduces an innovative path planning strategy tailored to Unmanned Aerial Manipulation (UAM) systems for pick-and-place applications. It offers a solution for effectively coordinating the aerial platform with the manipulator robot to navigate around obstacles. This is achieved by establishing two distinct paths, one for the aerial platform and another for the manipulator robot, ensuring seamless operation and preventing collisions within the entire system. The core of this approach relies heavily on the novel Random Geometric Model (RGM) to position the aerial platform close to the target object, thus confirming that the object is within reach of its end-effector. It employs sampling-based methods and geometric models to efficiently explore the configuration space and generate collision-free path. The fundamental concept underlying this approach is to randomly generate points within the workspace of the UAM as destination objectives. These points act as guiding references for the sampling-based algorithms, simplifying the creation of connectivity graphs between the starting position and the intended destinations for the aerial manipulator. The effectiveness of this strategy is demonstrated across various scenarios, highlighting its superior performance in terms of efficiency, computation time, and path length when compared to existing path planning techniques like RRT* and Informed RRT*.

Keywords: Path planning; sampling-based methods; pick-and-place; random geometric model; UAMs; UAV manipulator.

1. Introduction

Unmanned Aerial Vehicles (UAVs) have captured substantial attention in recent years, owing to their versatile applications, encompassing surveillance, reconnaissance, logistics, and gas distribution mapping applications [2]. However, despite these advancements, many UAVs face limitations in their ability to directly interact with and modify their environment. The integration of manipulator arms with UAV platforms has emerged as a transformative solution, unlocking new possibilities for intricate aerial tasks, commonly referred to as aerial manipulation [20].

Unmanned aerial manipulators boast enhanced capabilities, making them applicable in a wide array of scenarios. These applications encompass transporting vital first aid kits to remote and perilous locations where manned missions pose significant risks. They are invaluable for high-risk tasks such as power line maintenance, construction in otherwise inaccessible sites, and precise aerial pick-and-place operations, as evidenced in [7, 14, 25]. Additionally, they excel in scenarios necessitating manipulation and evacuation in hazardous environments, such as post-nuclear disaster situations or the delicate task of diffusing improvised explosive devices (IEDs) during military operations [23], all without jeopardizing human lives [19].

Received 18 January 2024; Accepted 30 October 2024; Published 9 January 2025. This paper was recommended for publication in its revised form by editorial board member, Shiyu Zhao.
Email Address: **zamoumhousseyn@gmail.com*

In the majority of these operations, aerial manipulators offer distinct advantages when compared to ground-based platforms. Their maneuverability allows them to navigate obstacles and rugged terrain with ease, all while maintaining the capability to impact their environment similarly to ground-based vehicles [9].

Aerial manipulation operations can be divided into three distinct phases: first, navigating from the take-off position to the vicinity of the workspace, ensuring collision avoidance and reactivity to unexpected obstacles; second, approaching the desired operation position with greater accuracy; and third, performing manipulation tasks with precise position and interaction control [21].

Many practical approaches for path planning of aerial manipulators decouple the planning of the aerial platform and the manipulator. Initially, these algorithms plan a trajectory for the center of gravity (CoG) of the aerial platform to position it suitably for manipulation. Subsequently, they plan the motion of the end-effector, assuming the aerial platform remains in the same position and orientation during manipulation. However, these approaches have several drawbacks. First, the geometric approach does not account for orientation, which is crucial in aerial manipulation. Second, the decoupling between the aerial platform and the manipulator can be inefficient in terms of energy and execution time. Third, this decoupling can become unfeasible when the manipulator's motion is necessary to avoid obstacles. Lastly, the decoupling is also impractical when the manipulator's movement significantly impacts the platform's motion [21].

Kinodynamic planning addresses the kinematic and dynamic constraints of aerial manipulators. A straightforward decoupling approach starts with geometric planning for a sphere surrounding the aerial platform, using methods like RRT to compute segments with zero velocity and acceleration at endpoints. This path is then transformed into a trajectory that satisfies the kinematic and dynamic constraints. Finally, velocities and accelerations are adjusted along local paths without altering their geometry, ensuring collision avoidance [1]. Control-aware planning, as proposed in, maintains surface contact, and some methods integrate motion planning and control to sustain this contact [26]. In the AEROARMS project, an RRT$*$ algorithm was used for simultaneous motion planning of the aerial platform and arm joints while transporting a bar in an obstacle-dense environment. An eight-variable planar model was employed for the aerial manipulator. The motion of the long-reach aerial manipulator's arms, when holding the bar, affects the aerial robot's CoG and subsequently its platform's motion. Without considering this coupling, collisions may occur. Therefore, an RRT$*$ algorithm with dynamic awareness was developed and implemented to avoid such collisions [5].

However, the development of an optimal, collision-free path for aerial manipulators to perform complex tasks with precision is one of the main challenges driven by several factors. Aerial Manipulators, unlike traditional UAVs, are comprised of both an aerial platform and a manipulator robot. This configuration introduces a significant obstacle avoidance challenge, necessitating the establishment of dual paths. These paths are vital; one guides the UAV, and the other guides the manipulator robot. Their purpose is to ensure seamless operation and prevent collisions throughout the entire system.

Furthermore, UAMs are classified as floating-base systems, distinct from fixed-base manipulators. This classification adds complexity, as achieving precise positioning of the aerial platform in close proximity to a target object and ensuring accessibility by the manipulator's end-effector demands a high level of precision. Addressing these challenges calls for the definition of specific mathematical tools to empower the aerial manipulator in fulfilling its designated tasks effectively.

Recently, there has been a growing interest in the use of sampling-based algorithms to solve the problem of path planning for aerial manipulators, such as RRT* and PRM* [15], Informed RRT*[11, 18], Fast Marching Trees (FMT*) in [13], C-FOREST in [22], and Batch Informed Trees (BIT*) developed in [10] are typically more efficient and scalable than other Graph-based searches path-planning algorithms [30] (such as Dijkstras algorithm [8] and A* [24]).

In their work, Brunner *et al.* [4] 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 [28], Yavari *et al.* 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 pre-grasp state while minimizing the motion of the arm. Limited-range sensors onboard facilitate on-the-fly planning using Machine-Learning based.

The authors in [16] 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 manner and incorporates an extra merging process to integrate the trees from the start and goal points. In [6], 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 [17], 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 *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 [29].

This paper presents a solution of path planning for Aerial Manipulators that leverages sampling-based algorithms in conjunction with Direct Geometric Model (DGM) and Inverse Geometric Model (IGM) equations to capture the interconnected motions between the UAV and its manipulator arm. The proposed methodology combines the advantages of sampling-based techniques and Random Geometric Model (RGM) methods, enabling the generation of UAV arrival points near the target object while ensuring the manipulator arm's reachability to the object. These algorithms rely on a meticulous representation of the robot's physical structure, encompassing factors like shape, size, joint angles, and boundaries. By leveraging this detailed information, these algorithms are capable of calculating feasible and safe paths while avoiding singularities or "unreachable spaces." These unreachable spaces refer to states wherein an aerial manipulator cannot manipulate an object due to limitations imposed by its physical structure or task-specific constraints.

Moreover, the proposed approach guarantees collision-free path not only for the UAV but also for the manipulator arm. Furthermore, this algorithm effectively explores high-dimensional spaces and offers efficient solutions in terms of computational time and path length, surpassing conventional methods reported in the existing literature. These features render it highly valuable in real-world scenarios that require precise and efficient path planning.

The remainder of the paper is structured as follows. Section 2 delves into the geometric modeling of aerial manipulator systems, including both the direct and IGMs of UAMs. Section 3 formulates the mathematical aspects of path planning for this category of systems. It also provides necessary definitions for key concepts in path planning. Section 4 thoroughly expounds the proposed path planning strategy and provides in-depth insight into the algorithm developed in this study. Section 5 presents the diverse results achieved when implementing the algorithm in various scenarios. Finally, Sec. 6 concludes by summarizing the results and offering recommendations for future research.

2. Geometric Modeling

We consider a UAM, which is a combined system of a UAV and an n-DoF robotic arm. The earth fixed frame $\{\mathcal{I}\} = (O_{\mathcal{I}}, X_{\mathcal{I}}, Y_{\mathcal{I}}, Z_{\mathcal{I}})$, the body fixe frame $\{\mathcal{B}\} = (O_{\mathcal{B}}, X_{\mathcal{B}}, Y_{\mathcal{B}}, Z_{\mathcal{B}})$ attached to the center of mass of UAV, the $\{\mathcal{B}_j\} = (O_j, X_j, Y_j, Z_j)$ reference frame that are attached to the link j of the arm, and the $\{\mathcal{B}_e\} = (O_e, X_e, Y_e, Z_e)$ reference frame, which is attached to the end effector. They are defined as in Fig. 1.

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

The first vector $\chi = [X_e^T \Phi_e^T]^T$, where $\Phi_e = [\phi_e \theta_e \psi_e]^T$ is expressed as the generalized coordinate of the end effector.

The second vector represented by the collection of state variables of the UAV and the arm joint as $\xi = [X_{\mathcal{B}}^T\ \Phi_{\mathcal{B}}^T\ \mathcal{Q}_a^T]^T$, with $\Phi_{\mathcal{B}} = [\phi\ \theta\ \psi]^T$ gives the orientation of the UAV with respect to the fixed reference frame $\{\mathcal{I}\}$, $\mathcal{Q}_a = [q_1\ q_2\ \ldots\ q_n]^T$ represent the joint vector of the n-DoF manipulator's.

2.1. *Direct geometrical models*

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

Definition 1. Consider $\mathcal{D}_{\mathcal{G}}$ as a mathematical function describing the transition from the joint space $\mathcal{N} \subseteq \mathbb{R}^{6+n}$ to the operational coordinate space $\mathcal{M} \subseteq \mathbb{R}^6$ as follows:

$$\begin{aligned} &\mathcal{D}_{\mathcal{G}} : \mathcal{N} \to \mathcal{M} \\ &\xi \to \chi = \mathcal{D}_{\mathcal{G}}(\xi) \end{aligned}. \tag{1}$$

The $\mathcal{D}_{\mathcal{G}}$ function is derived geometrically by solving the following system of equations:

$$\begin{cases} {}^{\mathcal{I}}P_e = {}^{\mathcal{I}}P_{\mathcal{B}} + {}^{\mathcal{I}}\mathcal{R}_{\mathcal{B}}(\Phi_{\mathcal{B}})\,{}^{\mathcal{B}}P_e(\mathcal{Q}_a) \\ {}^{\mathcal{I}}\mathcal{R}_e(\Phi_e) = {}^{\mathcal{I}}\mathcal{R}_{\mathcal{B}}(\Phi_{\mathcal{B}})\,{}^{\mathcal{B}}\mathcal{R}_e(\mathcal{Q}_a) = \mathcal{R}(\Phi_{\mathcal{B}}, \mathcal{Q}_a), \end{cases} \tag{2}$$

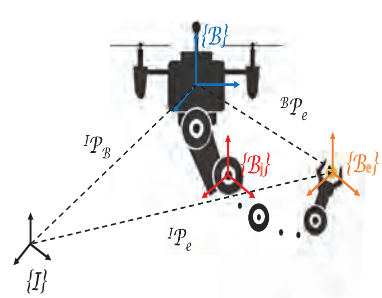


Fig. 1. Configuration space of an unmanned aerial manipulator (i.e. a UAV with a robot arm with n-DoF).

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

Based on the system of Eq. (2), the function $\mathcal{D}_{\mathcal{G}}$ can be given as follows:

$$\mathcal{D}_{\mathcal{G}} = \begin{bmatrix} \mathcal{D}_{\mathcal{G}}^{+} \\ \mathcal{D}_{\mathcal{G}}^{-} \end{bmatrix} = \begin{bmatrix} X_e \\ \Phi_e \end{bmatrix} = \begin{bmatrix} X_{\mathcal{B}} + \mathcal{G}(\Phi_{\mathcal{B}}, \mathcal{Q}_a) \\ \mathcal{H}(\Phi_{\mathcal{B}}, \mathcal{Q}_a) \end{bmatrix}, \tag{3}$$

where function $\mathcal{G}(\Phi_{\mathcal{B}}, \mathcal{Q}_a) = {}^{\mathcal{I}}\mathcal{R}_{\mathcal{B}}(\Phi_{\mathcal{B}})\, {}^{\mathcal{B}}P_e(\mathcal{Q}_a)$ and the function $\mathcal{H}(\Phi_{\mathcal{B}}, \mathcal{Q}_a)$ is defined as

$$\mathcal{H}(\Phi_{\mathcal{B},q}) = \begin{cases} \operatorname{atan2}(\mathcal{R}(\Phi_{\mathcal{B}}, q)_{23}, \mathcal{R}(\Phi_{\mathcal{B}}, q)_{13}) \\ \operatorname{atan2}(\sqrt{\mathcal{R}_{13}^2 + \mathcal{R}_{23}^2}, -\mathcal{R}(\Phi_{\mathcal{B}}, q)_{33}) \\ \operatorname{atan2}(\mathcal{R}(\Phi_{\mathcal{B}}, q)_{32}, -\mathcal{R}(\Phi_{\mathcal{B}}, q)_{31}), \end{cases} \tag{4}$$

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

2.2. *Inverse geometrical models*

The inverse geometric approach for UAM is used to calculate the UAV position and manipulator arm joint variables required to move the end effector to a desired position.

Definition 2. Consider $\mathcal{I}_{\mathcal{G}}$ as a mathematical function describing the transition from the operational coordinate space $\mathcal{M}$ to the joint space $\mathcal{N}$ as follows:

$$\begin{aligned} &\mathcal{I}_{\mathcal{G}} : \mathcal{M} \to \mathcal{N} \\ &\chi \to \xi = \mathcal{I}_{\mathcal{G}}(\chi). \end{aligned} \tag{5}$$

Let's consider Eq. (3). We have a system of nonlinear 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, several solutions can be found, as well as no solutions can be found after an analytical mathematical calculation.

Several numerical and analytical approaches have been proposed to solve this problem. Among these approaches, the general inversion method of Paul in [27] emerges as a leading mathematical technique used to solve inverse geometric problems in robotics. This approach uses homogeneous transformation matrices ${}^{j}T_i$ to achieve the desired position and orientation of the end effector given by the matrix $\mathcal{U}({}^{\mathcal{I}}\mathcal{R}_e(\Phi_e), {}^{\mathcal{I}}P_e)$ as follows:

$$\mathcal{U}({}^{\mathcal{I}}\mathcal{R}_e(\Phi_e), {}^{\mathcal{I}}P_e) = \begin{pmatrix} {}^{\mathcal{I}}\mathcal{R}_e(\Phi_e) & {}^{\mathcal{I}}P_e \\ 0_{1\times 3} & 1 \end{pmatrix}. \tag{6}$$

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

$$\mathcal{U} = {}^{\mathcal{I}}T_e = {}^{\mathcal{I}}T_{\mathcal{B}}\, {}^{\mathcal{B}}T_1\, {}^{1}T_2\, {}^{2}T_3 \ldots {}^{n}T_e \tag{7}$$

we use this Eq. (7) to compute the manipulator arm state variables $\mathcal{Q}_a$ using the following recursive method:

$$\left.\begin{aligned} &\mathcal{U} = {}^{\mathcal{I}}T_{\mathcal{B}}\, {}^{\mathcal{B}}T_1\, {}^{1}T_2\, {}^{2}T_3 \ldots {}^{n}T_e \\ &{}^{\mathcal{B}}T_{\mathcal{I}}\mathcal{U} = {}^{\mathcal{B}}T_1\, {}^{1}T_2\, {}^{2}T_3 \ldots {}^{n}T_e \\ &{}^{1}T_{\mathcal{B}}\, {}^{\mathcal{B}}T_{\mathcal{I}}\mathcal{U} = {}^{1}T_2\, {}^{2}T_3 \ldots {}^{n}T_e \\ &{}^{2}T_1\, {}^{1}T_{\mathcal{B}}\, {}^{\mathcal{B}}T_{\mathcal{I}}\mathcal{U} = {}^{2}T_3 \ldots {}^{n}T_e \\ &\vdots \\ &{}^{n}T_{n-1} \ldots {}^{2}T_1\, {}^{1}T_{\mathcal{B}}\, {}^{\mathcal{B}}T_{\mathcal{I}}\mathcal{U} = {}^{n}T_e \end{aligned}\right\} \to \mathcal{Q}_a. \tag{8}$$

With the assumption of hovering manipulation ($\phi = \theta = 0$), we can easily deduce the vector $\mathcal{Q}_a = \mathcal{L}(X_e, \Phi_e)$ by solving the system of Eq. (8). 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 Eq. (3), hence the position expression of the UAV is $X_{\mathcal{B}} = X_e - \mathcal{G}(\Phi_{\mathcal{B}}, \mathcal{Q}_a)$.

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

$$\mathcal{I}_{\mathcal{G}} = \begin{bmatrix} \mathcal{I}_{\mathcal{G}}^{+} \\ \Phi_{\mathcal{B}} \\ \mathcal{I}_{\mathcal{G}}^{-} \end{bmatrix} = \begin{bmatrix} X_{\mathcal{B}} \\ \Phi_{\mathcal{B}}^{0} \\ \mathcal{Q}_a \end{bmatrix} = \begin{bmatrix} X_e - \mathcal{G}(\Phi_{\mathcal{B}}^{0}, \mathcal{Q}_a) \\ \Phi_{\mathcal{B}}(0, 0, \psi_d) \\ \mathcal{L}(X_e, \Phi_e) \end{bmatrix}. \tag{9}$$

2.3. *Application example*

The direct and IGMs can be derived through various methods. In this work, we opt for the modified Denavit–Hartenberg convention, often referred to as the Khalil–Kleinfinger convention. This method is chosen for its simplicity and ease of implementation.

Consider our aerial manipulation system developed in [12] (shown in Fig. 2). The manipulator's arm consists of three main rigid links which represent the body of the arm and three rotary joints capable of three rotations $(\theta_1, \theta_2, \theta_3)$. Figure 3 shows the final version of the prototype designed by Solidworks.

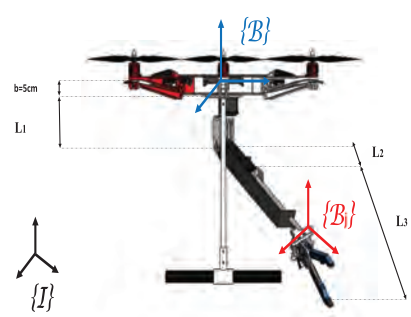


Fig. 2. Visual diagram depicting the system-related frame.

Fig. 3. Final structure of the manipulator arm designed by Solidworks.

The transformation matrix from the reference frame $\{\mathcal{B}_j\} = (O_j, X_j, Y_j, Z_j)$ to the reference frame $\{\mathcal{B}_i\}$ is given by the following formula:

$$ {}^iT_j = \begin{bmatrix} {}^i\mathcal{R}_j & {}^iP_j \\ 0_{1\times3} & 1 \end{bmatrix}. \tag{10}$$

The four M.D.H. parameters are used to express the rotation matrix ${}^i\mathcal{R}_j$ as follows:

$$ {}^i\mathcal{R}_j = \begin{pmatrix} c_{\theta_j} & -s_{\theta_j} & 0 \\ c_{\alpha_j}s_{\theta_j} & c_{\alpha_j}c_{\theta_j} & -s_{\alpha_j} \\ s_{\alpha_j}s_{\theta_j} & s_{\alpha_j}c_{\theta_j} & c_{\alpha_j} \end{pmatrix} \tag{11}$$

and the position vector:

$$ {}^iP_j = \begin{pmatrix} d_j \\ -r_js_{\alpha_j} \\ r_jc_{\alpha_j} \end{pmatrix} \tag{12}$$

Based on the M.D.H. parameters for the 3-link manipulator presented in Table 1, the five transformation matrices are given as follows:

$$ {}^\mathcal{B}T_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -b \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^\mathcal{B}\mathcal{R}_0 & {}^\mathcal{B}P_0 \\ 0_{1\times3} & 1 \end{bmatrix}, \tag{13}$$

$$ {}^0T_1 = \begin{bmatrix} c_{\theta_1} & -s_{\theta_1} & 0 & 0 \\ s_{\theta_1} & c_{\theta_1} & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^0\mathcal{R}_1 & {}^0P_1 \\ 0_{1\times3} & 1 \end{bmatrix}, \tag{14}$$

$$ {}^1T_2 = \begin{bmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_{\theta_2} & c_{\theta_2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^1\mathcal{R}_2 & {}^1P_2 \\ 0_{1\times3} & 1 \end{bmatrix}, \tag{15}$$

Table 1. M.D.H. parameter table.

j	σ_i	d_j	α_j	r_j	θ_j
1	0	0	0	L_1	θ_1
2	0	0	$+\frac{\pi}{2}$	0	θ_2
3	0	0	$-\frac{\pi}{2}$	L_2	θ_3
e	/	0	0	L_3	0

$$ {}^2T_3 = \begin{bmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ -s_{\theta_3} & -c_{\theta_3} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^2\mathcal{R}_3 & {}^2P_3 \\ 0_{1\times3} & 1 \end{bmatrix}, \tag{16}$$

$$ {}^3T_e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^3\mathcal{R}_e & {}^3P_e \\ 0_{1\times3} & 1 \end{bmatrix} \tag{17}$$

the passage from the body frame $\{\mathcal{B}\}$ to the reference frame attached to the terminal organ $\{\mathcal{B}_e\}$, is provided by the matrix:

$$ {}^\mathcal{B}T_e = {}^\mathcal{B}T_0\,{}^0T_1\,{}^1T_2\,{}^2T_3\,{}^3T_e = \begin{bmatrix} {}^\mathcal{B}\mathcal{R}_e & {}^\mathcal{B}P_e \\ 0_{1\times3} & 1 \end{bmatrix} \tag{18}$$

${}^\mathcal{B}\mathcal{R}_e$ is the rotation matrix from frame $\{\mathcal{B}_e\}$ to frame $\{\mathcal{B}\}$, is given by

$$ {}^\mathcal{B}\mathcal{R}_e = \begin{pmatrix} c_{\theta_1}c_{\theta_2}c_{\theta_3} - s_{\theta_1}s_{\theta_3} & -c_{\theta_1}c_{\theta_2}s_{\theta_3} - s_{\theta_1}c_{\theta_3} & -c_{\theta_1}s_{\theta_2} \\ c_{\theta_2}c_{\theta_3}s_{\theta_1} + c_{\theta_1}s_{\theta_3} & s_{\theta_1}s_{\theta_3}c_{\theta_2} + c_{\theta_1}c_{\theta_3} & -s_{\theta_1}s_{2\theta_2} \\ s_{\theta_2}c_{\theta_3} & -s_{\theta_2}s_{\theta_3} & c_{\theta_2} \end{pmatrix} \tag{19}$$

${}^\mathcal{B}P_e$ is the position of the end effector relative to the reference frame $\{\mathcal{B}\}$

$$ {}^\mathcal{B}P_e = \begin{pmatrix} {}^\mathcal{B}x_e \\ {}^\mathcal{B}y_e \\ {}^\mathcal{B}z_e \end{pmatrix} = \begin{pmatrix} -c_{\theta_1}s_{\theta_2}(L_2+L_3) \\ s_{\theta_1}s_{\theta_2}(L_2+L_3) \\ -c_2(L_2+L_3) - L_1 - b \end{pmatrix} \tag{20}$$

We can define the position of the end effector as follows:

$$ {}^\mathcal{I}P_e = {}^\mathcal{I}P_\mathcal{B} + {}^\mathcal{I}\mathcal{R}_\mathcal{B}\,{}^\mathcal{B}P_e \tag{21}$$

with ${}^\mathcal{I}\mathcal{R}_\mathcal{B}$ given as

$$ {}^\mathcal{I}\mathcal{R}_\mathcal{B} = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix}. \tag{22}$$

Equation (21) can be written as

$$ \begin{pmatrix} x_e \\ x_e \\ z_e \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + {}^\mathcal{I}\mathcal{R}_\mathcal{B} \begin{pmatrix} -c_1s_2(L_2+L_3) \\ s_1s_2(L_2+L_3) \\ -c_2(L_2+L_3) - L_1 - b \end{pmatrix} \tag{23}$$

Based on the system of Eq. (22), the function $\mathcal{D}_{\mathcal{G}}^{+}$ can be given as follows:

$$\mathcal{D}_{\mathcal{G}}^{+} = X_e = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + {}^{\mathcal{I}}\mathcal{R}_{\mathcal{B}} \begin{pmatrix} -c_1 s_2(L_2 + L_3) \\ s_1 s_2 (L_2 + L_3) \\ -c_2(L_2 + L_3) - L_1 - b \end{pmatrix} \quad (24)$$

To address the direct geometric problem, it's essential to calculate the d function, which corresponds to the set of Euler angles that align with the rotation matrix of the end-effector. Let us base on the calculation of matrix ${}^{\mathcal{I}}\mathcal{R}_{\mathcal{B}}$, which can be given as follows:

$${}^{\mathcal{I}}\mathcal{R}_e = {}^{\mathcal{I}}\mathcal{R}_{\mathcal{B}}\,{}^{\mathcal{B}}\mathcal{R}_e = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}. \quad (25)$$

When the terms r_{ij} are function of q and $\Phi_{\mathcal{B}}$, we assume that elements r_{13} and r_{33} are not equal to zero, we can express it as follows:

$$\begin{cases} \phi_e = \operatorname{atan2}(r_{23}, r_{13}) \\ \theta_e = \operatorname{atan2}(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}) \\ \psi_e = \operatorname{atan2}(r_{32}, -r_{31}) \end{cases} \quad (26)$$

Choosing a positive sign for the term $\sqrt{r_{13}^2 + r_{23}^2}$ limits the range of possible values of θ_e to $[0, \pi]$. So, the function $\mathcal{D}_{\mathcal{G}}^{-}$ is defined as

$$\mathcal{D}_{\mathcal{G}}^{-} = \Phi_e \mathcal{H}(\Phi_{\mathcal{B}}, q) = \begin{pmatrix} \operatorname{atan2}(r_{23}, r_{13}) \\ \operatorname{atan2}(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}) \\ \operatorname{atan2}(r_{32}, -r_{31}) \end{pmatrix}. \quad (27)$$

Now, let's turn our attention to the IGM, which involves expressing the joint coordinates in terms of the operational coordinates. Solving the geometric inverse model is a complex task for several reasons:

(1) The problem at hand typically involves a system of nonlinear equations.
(2) Multiple solutions may exist for a single problem.
(3) In the case of redundant systems, there could be an infinite number of potential solutions.
(4) Analytical mathematical solutions may not be readily available.

However, when dealing with redundant manipulators, it's possible to find multiple potential solutions for a given end-effector position. we denote by ${}^{\mathcal{I}}\mathcal{P}_e^d$ and ${}^{\mathcal{I}}\mathcal{R}_e^d$, the desired end-member position and orientation, expressed by

$${}^{\mathcal{I}}\mathcal{P}_e = \begin{bmatrix} x_e^d & y_e^d & z_e^d \end{bmatrix}^T, \quad (28)$$

$${}^{\mathcal{I}}\mathcal{R}_e^d = \begin{pmatrix} r_x & n_x & a_x \\ r_y & n_y & a_y \\ r_z & n_z & a_z \end{pmatrix}. \quad (29)$$

Let $\mathcal{U}_0$ be the transformation matrix describing the desired orientation and position:

$$\mathcal{U}_0 = \begin{pmatrix} {}^{\mathcal{I}}\mathcal{R}_e^d & {}^{\mathcal{I}}P_e^d \\ 0_{1\times 3} & 1 \end{pmatrix}. \quad (30)$$

To find the joint values for a given end-member position, the UAM is in the situation of gripping and manipulating the target. In a situation of flight stability, i.e. its speed is close to zero. In this case, the pitch and roll angles are close to zero.

2.3.1. *Assumptions*

In a hovering position of the UAM, we assume that:

- The pitch and roll angles equal to zero ($\phi = \theta = 0$), no movement on x and y axes.
- $\forall {}^{\mathcal{I}}P_e^d \in \mathbb{R}^3$ the UAM can reach any point in space.

This means that roll and pitch angles are virtually zero, so we'll set them to zero, from (7) and (8), the inverse geometry of the system can be derived from the following equation:

$$\mathcal{U}_0 = {}^{\mathcal{I}}T_e = {}^{\mathcal{I}}T_0\,{}^{0}T_1\,{}^{1}T_2\,{}^{2}T_3\,{}^{3}T_e. \quad (31)$$

The three variables θ_1, θ_2 and θ_3 expressed as follows (we can suppose any value for ψ and obtain θ_1):

$$\begin{cases} \theta_1 + \psi = \operatorname{atan2}(-p_y, p_x) \\ \theta_2 = \operatorname{atan2}(c_1 p_x - s_1 p_y, s_1 p_x + c_1 p_y) \\ \theta_3 = \operatorname{atan2}(s_1 r_x + c_1 r_y, s_1 n_x + c_1 n_y). \end{cases} \quad (32)$$

Therefore, the function $\mathcal{I}_{\mathcal{G}}^{-}$ described the variable q (notic$\Phi_{\mathcal{B}}^0 = \Phi_{\mathcal{B}}(0, 0, \psi_d)$) can be given as follows:

$$\mathcal{I}_{\mathcal{G}}^{-} = q = \begin{pmatrix} \operatorname{atan2}(-p_y, p_x) - \psi_d \\ \operatorname{atan2}(c_1 p_x - s_1 p_y, s_1 p_x + c_1 p_y) \\ \operatorname{atan2}(s_1 r_x + c_1 r_y, s_1 n_x + c_1 n_y) \end{pmatrix}. \quad (33)$$

Finally, the UAV position is determined from Eq. (23) as follows:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_e^d \\ y_e^d \\ z_e^d \end{pmatrix} - \begin{pmatrix} -c_{\theta_1+\psi} s_{\theta_2}(L_2 + L_3) \\ s_{\theta_1+\psi} s_{\theta_2}(L_2 + L_3) \\ -c_{\theta_2}(L_2 + L_3) - L_1 - b \end{pmatrix}. \quad (34)$$

The function $\mathcal{I}_{\mathcal{G}}^{+}$ described the IGM can be given as follows:

$$\mathcal{I}_{\mathcal{G}}^{+} = X_{\mathcal{B}} = \begin{pmatrix} x_e^d + c_{\theta_1+\psi} s_{\theta_2}(L_2 + L_3) \\ y_e^d - s_{\theta_1+\psi} s_{\theta_2}(L_2 + L_3) \\ z_e^d + c_{\theta_2}(L_2 + L_3) + L_1 + b \end{pmatrix}. \quad (35)$$

3. Mathematical Formulations of the Problem

Path planning for an aerial manipulator involves identifying a collision-free path that both the UAV and its manipulator arm can follow, starting from an initial position and reaching multiple points in the vicinity of the target object. The primary objective is to guarantee that the manipulator arm successfully reaches the target object while avoiding any potential collisions.

The formulation of this problem may vary depending on the scenario and the scenario objectives, but here is the mathematical formulation used in path planning to pick up an object:

- Let $X \subseteq \mathbb{R}^3$ be 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", states in the vicinity of the target object in which an aerial manipulator cannot pick up an object it is unable to reach due to the limitations of its physical structure or the constraints of the task.
- Let $X_{\text{free}} = X \backslash X_{\text{obs}} \cap X \backslash X_{\text{dead}}$ be 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}.

3.1. *Point-to-point problem*

Most of the existing path planning strategies aim to find a feasible and optimal path between a starting point and an end point (i.e. a Point-To-Point (PTP) problem). However, some robotic applications, such as pick-up and place scenarios, require other types of path planning techniques involving what we call a Point-To-PointS path planning problem (PTPS). In other words, the UAV manipulator has several arrival points close to the target object, in order to its end effector be able to reach the target object.

Let $X_{\mathcal{B},\text{start}}$ and $X_{e,\text{start}}$ represent the initial positions of the UAV and its manipulator arm, respectively. $X_{\text{final}} \in \mathbb{G}$ denotes the ultimate position of the UAV in the vicinity of the target object located at position X_{goal}, where $\mathbb{G} = \{X_{i|i=0,\ldots m} | \exists \mathcal{Q}_a \in \mathbb{M} : X_i = X_{\text{goal}} - \mathcal{G}(\Phi^0_{\mathcal{B}}, \mathcal{Q}_a)\}$ is the set of goal points chosen by the UAV, and $\mathbb{M} \subseteq \mathbb{R}^n$ is the space of joint variables $\mathcal{Q}_a$ boundary between $\mathcal{Q}_{\min}$ and $\mathcal{Q}_{\max}$, denote the minimum and maximum limits of joint, respectively.

Definition 3. 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:

$$\begin{aligned} \sigma &: [0,1] \times [0,1] \to X_{\text{free}} \times X_{\text{free}} \\ \lambda &\to \sigma(\lambda) = [\sigma_{\mathcal{B}}, \sigma_e] \end{aligned} \tag{36}$$

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

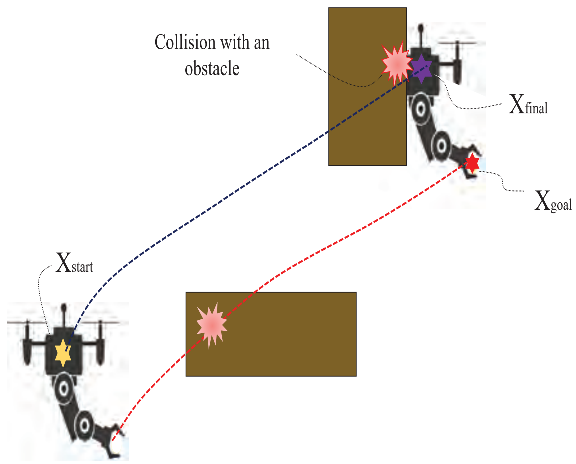


Fig. 4. UAV and end effector path obtained from the IGM.

The search for the optimal path σ_e^* and $\sigma_{\mathcal{B}}^*$, which minimizes a specified cost function, $c : \Gamma \to \mathbb{R}^+$, 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:

$$\sigma_{\mathcal{B}}^* = \arg\min_{\sigma_{\mathcal{B}} \in \Gamma} \{c(\sigma_{\mathcal{B}}) | \forall s \in [0,1], \sigma_{\mathcal{B}}(s) \in X_{\text{free}}\}. \tag{37}$$

One advantage of the geometric 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, as shown in Fig. 4.

In contrast, sampling-based methods such as RRT and RRT* are helpful for generating feasible paths that satisfy constraints such as obstacle avoidance, as shown in Fig. 5.

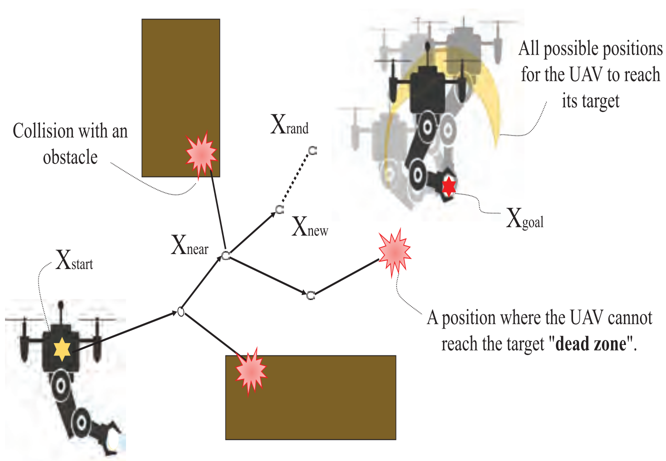


Fig. 5. The path planning through the sampling approaches.

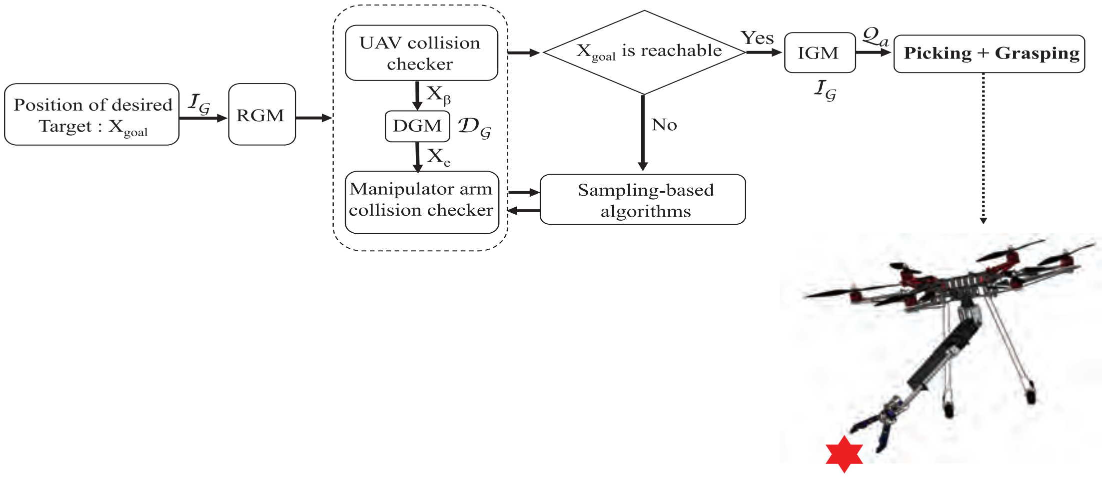


Fig. 6. Illustrative diagram of a UAV manipulator's path planning strategy for picking up an object.

The solution presented in this work comprises two distinct methods: a sampling-based approach and a geometrical model of the system (as illustrated in Fig. 6). Both methods consider the coordinated positioning of the UAV and its end effector to facilitate the efficient reaching of the target object. By combining these two techniques, the proposed solution aims to optimize the path planning process and enable the successful manipulation of the target object.

This work presents a solution of path planning for Aerial Manipulators that leverages sampling-based algorithms in conjunction with DGM and IGM equations to capture the interconnected motions between the UAV and its manipulator arm (while ensuring a safe transition between the DGM and the IGM using the functions $\mathcal{D}_g$ and $\mathcal{I}_g$ developed in Sec. 2). The following section presents the proposed algorithmic solution, all based on the geometric model developed in Sec. 2.

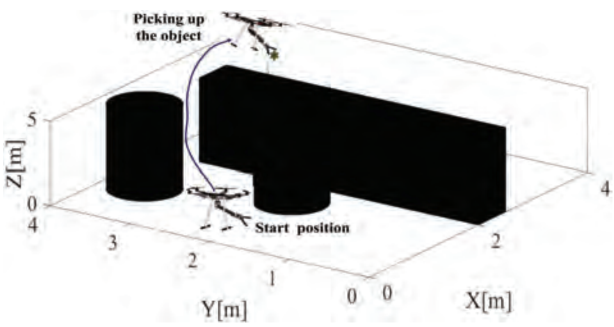
4. Path Planning Strategy

Sampling-based planners' algorithms have shown remarkable potential in efficiently solving path planning problems for high-dimensional robotic systems. These methods excel in scenarios where traditional approaches struggle due to the complexity of the configuration space, enabling the generation of feasible motion paths with significantly reduced computational cost. A key advantage of some sampling-based planners, such as Informed RRT*, lies in their ability to not only find solutions but also optimize motion plans according to specific cost functions. By incorporating heuristics, these planners are guided toward more optimal regions of the search space, thereby improving both the quality and efficiency of the generated paths. Informed RRT*, for instance, leverages information about the problem's structure to refine its exploration, allowing it to focus on areas more likely to yield low-cost solutions. This capability makes sampling-based planners an attractive choice for applications requiring fast, optimized, and reliable path planning, especially in complex environments with stringent performance criteria. Ongoing research continues to explore enhancements to these algorithms, further pushing their effectiveness in aerial robotic systems.

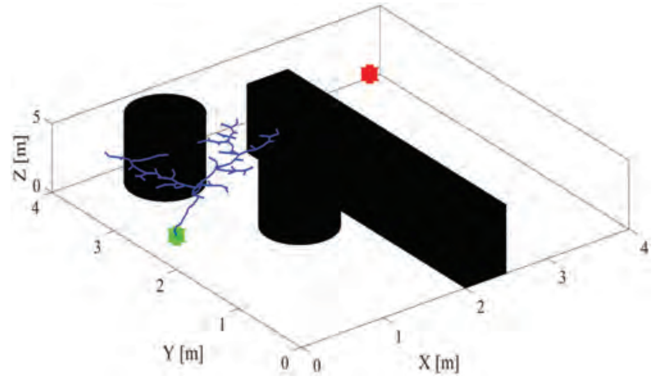
The algorithm is an extension of the RRT algorithm, which uses heuristics to guide the exploration of the RRT tree to the goal location using information about the distance to the goal and the quality of the paths that have been explored so far. These techniques allow Informed RRT* to find an optimal path to the goal in a more efficient manner than the original RRT algorithm.

In this work, path planning for a manipulator UAV using sampling and RGM-based methods is a technique used to generate feasible path for an UAV with a manipulator's arm. This technique is divided into two steps, in the first step the sampling is based on Informed RRT* to ensure optimal search in the working space where the target object is located while avoiding obstacles. In the second stage, 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 $\mathcal{I}_g$ developed in Sec. 2.

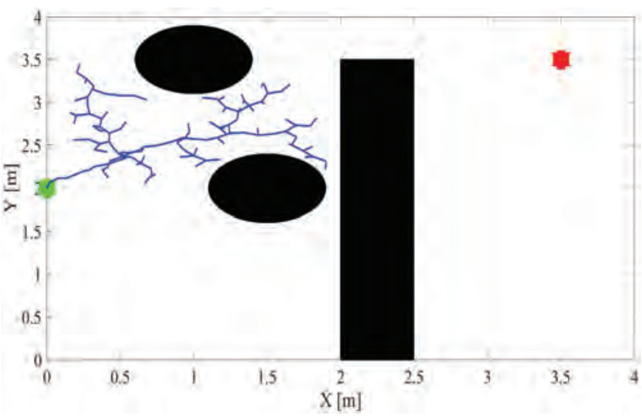
In the first phase (as shown in Figs. 7(b) and 7(c) 3D and 2D representation, respectively), the target object is indicated by a goal point X_{goal} in red and the arrival point X_{final} is shown in green. The blue paths represent the trajectories generated by RRT and informed RRT from the UAVs initial position, represented by a green star. These trajectories



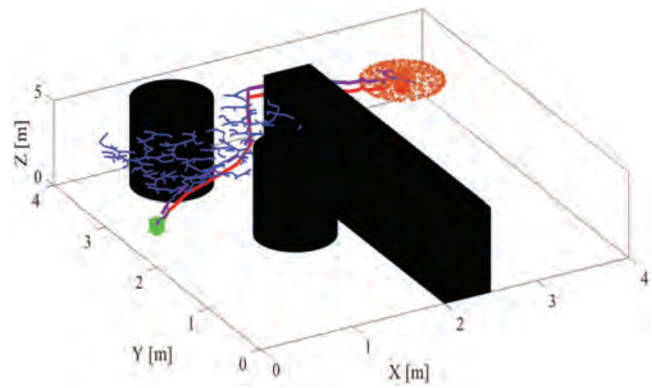
(a) The path planning phases of the aerial manipulator that must pass to pick up the object.



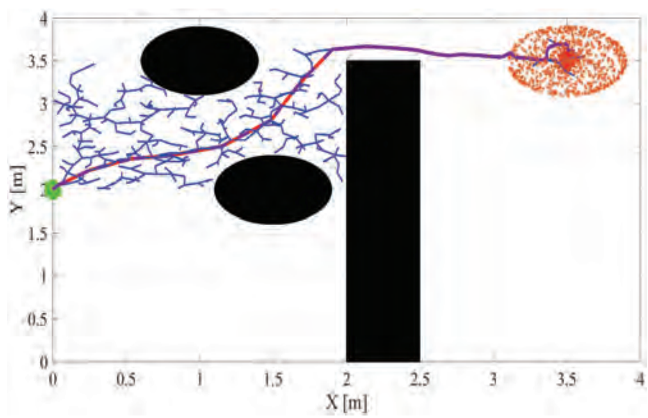
(b) During each iteration in the first phase, the search extends to the target object around the goal point using RGM.



(c) 2D representation of the first phase of the search for the target object.



(d) 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.



(e) 2D representation of the final phase to pick up the object.

Fig. 7. The start and end positions of the UAM are shown in green and red stars. Orange points indicate the final position of the UAV near the target object. Each purple and red line shows the path of UAV and its effector position, respectively.

expand iteratively, taking into account drone and manipulator constraints, to explore the space around the arrival point X_{final} and ensure that the manipulator arm can reach the target object located at position X_{goal}.

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 Figs. 7(d) and 7(e), which shows the

Algorithm 1. S-RGM*$(X_{\mathcal{B},\text{start}}, X_{e,\text{start}}, \mathcal{Q}_{a,\text{start}}, X_{\text{goal}}, \Phi_{e,\text{goal}})$

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

optimized final path, highlighting how the search process successfully navigates to the target object while maintaining collision-free conditions.

Algorithm 1 presents the pseudocode for the suggested algorithm using the RGM.

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

4.1. *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 $\Phi_{\mathcal{B}}^{0}$, $\mathcal{Q}_a$, all based on Eq. (38)

$$X_{\mathcal{B},\text{RGM}} = \mathcal{I}_{\mathcal{G}}^{+}(X_{\text{goal}}) = X_{\text{goal}} - \mathcal{G}(\Phi_{\mathcal{B},\text{rand}}^{0}, \mathcal{Q}_{a,\text{rand}}). \tag{38}$$

As shown in Eq. (12), the RGM uses random values of the articulation variables $\mathcal{Q}_a$ and the attitude of the UAM $\Phi_{\mathcal{B}}^{0}$ to obtain all the possible positions of the drone so that it can be manipulated with its arm correctly (Fig. 8 show the cloud of points generated by the RGM for two types of UAM, $\mathcal{H}$-RRR in [31] and $\mathcal{Q}$-PRR in [3]).

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 path to the target.

4.2. *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 $X_{\mathcal{B},\text{nearest}}$.

4.3. *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.

4.4. *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.

4.5. *Check collision with obstacle*

Obstacle avoidance for aerial manipulators involves ensuring that both the manipulator arm and the UAV can navigate

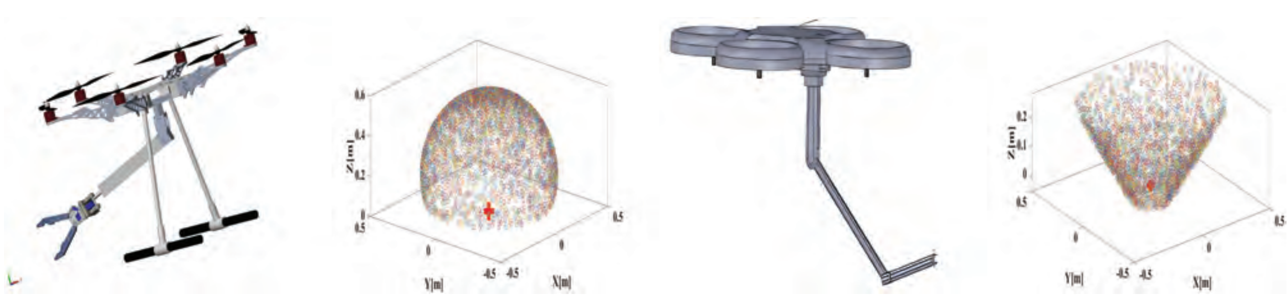


Fig. 8. RGM cloud points for two types of UAM, $\mathcal{H}$-RRR and $\mathcal{Q}$-PRR.

Algorithm 2. RGM(X_{goal})

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

through their environment without colliding with obstacles. Here are explanations for both cases:

(1) In the context of aerial manipulators, the manipulator arm's obstacle avoidance focuses on preventing collisions between the manipulator's links and any obstacles in its surroundings. The obstacle avoidance strategy for the manipulator arm may involve altering joint angles, adjusting the end-effector pose, or even temporarily halting motion until a safe trajectory can be planned.

(2) For the UAV component of the aerial manipulator, obstacle avoidance is vital to prevent collisions between the flying platform and any obstacles present in its flight path. UAV obstacle avoidance is typically achieved through the integration of sensors to perceive the environment. Algorithms, such as potential fields or rapidly exploring random trees (RRT), are commonly employed for UAV obstacle avoidance.

An advanced planning algorithm is proposed in this work to achieve efficient obstacle avoidance for both aspects of the aerial manipulator system by introducing new function of collision verification.

This function based on the transition between the DGM and the IGM using the functions $\mathcal{D}_g$ and $\mathcal{I}_g$ is developed in Sec. 2. It examines whether the drone's path from the nearest node $X_{\mathcal{B},\text{nearest}}$ in the configuration $X_{\mathcal{B},\text{rand}}$ to the new configuration $X_{\mathcal{B},\text{new}}$ remains free of collisions. Subsequently, the verification process assesses whether the path of the manipulator arm, spanning from the nearest node $X_{e,\text{nearest}}$ to the fresh configuration $X_{e,\text{new}}$, also maintains a collision-free status. This entails scrutinizing the absence of collisions between the aerial manipulator and the obstacles within the environment, as delineated by the map.

Introducing safety margins around boundary obstacles is crucial to provide a buffer zone for the aerial manipulator, ensuring that the manipulator maintains a safe distance from the boundaries to avoid collisions. These obstacles often present physical or geographical constraints that the aerial manipulator must navigate around or avoid altogether. The following illustration depicts how an aerial manipulator can successfully avoid boundary obstacles.

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

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

4.6. *Add function*

This function adds a new configuration $X_{\mathcal{B},\text{new}}$ to the tree $\mathcal{T}$ by connecting it to its nearest neighbor $X_{\mathcal{B},\text{nearest}}$ 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 $X_{\mathcal{B},\text{near}}$.

4.7. *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.

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

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

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

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

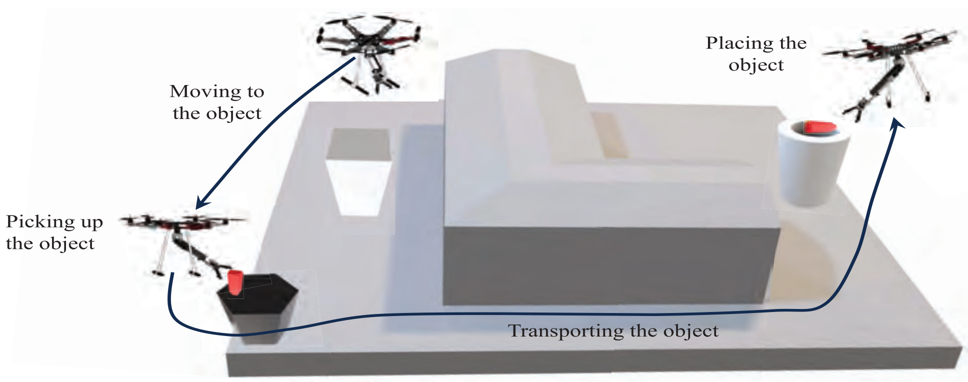
4.8. *DualPath function*

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.

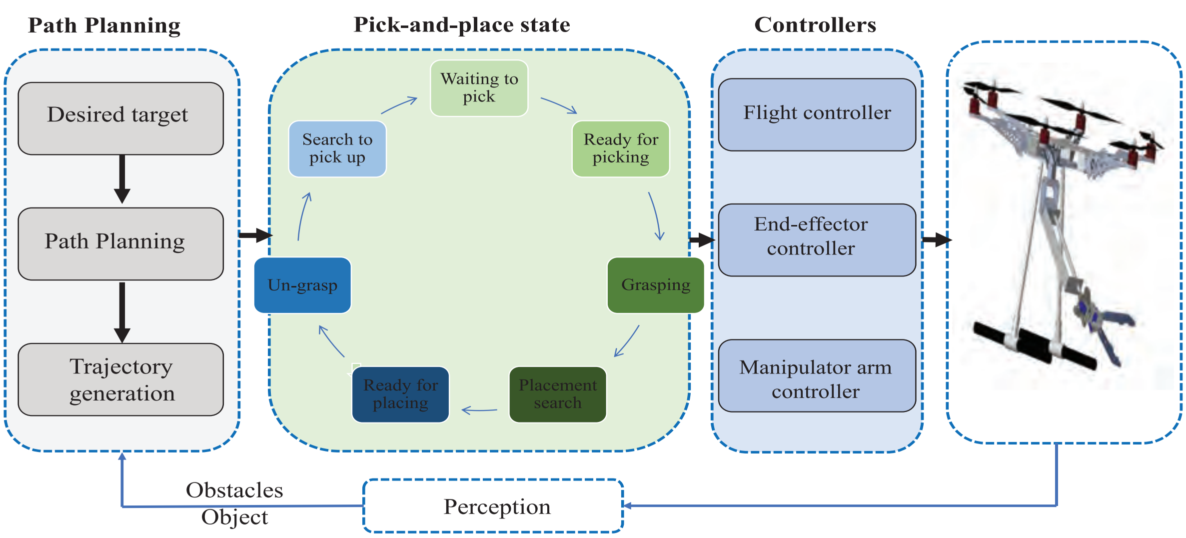
4.9. *Aerial pick and place*

Aerial pick-and-place path planning refers to the process of generating a path or route for an aerial vehicle, such as a UAV, to autonomously navigate and manipulate objects by picking them up from one location and placing them at another designated location, as shown in Fig. 9(a).

This involves a series of complex steps that allow the drone to interact with its environment in a coordinated and precise manner. Figure 9(b) depicts the comprehensive architecture of the aerial manipulator planning and control system designed for the task of picking up and placing objects. This architecture is organized into three distinct components: path planning, pick and place state, and controller.



(a) Aerial manipulation for pick-and-place operations using an aerial vehicle type hexarotor.



(b) Overall structure of the proposed method for aerial pick-and-place.

Fig. 9. Aerial pick-and-place illustration.

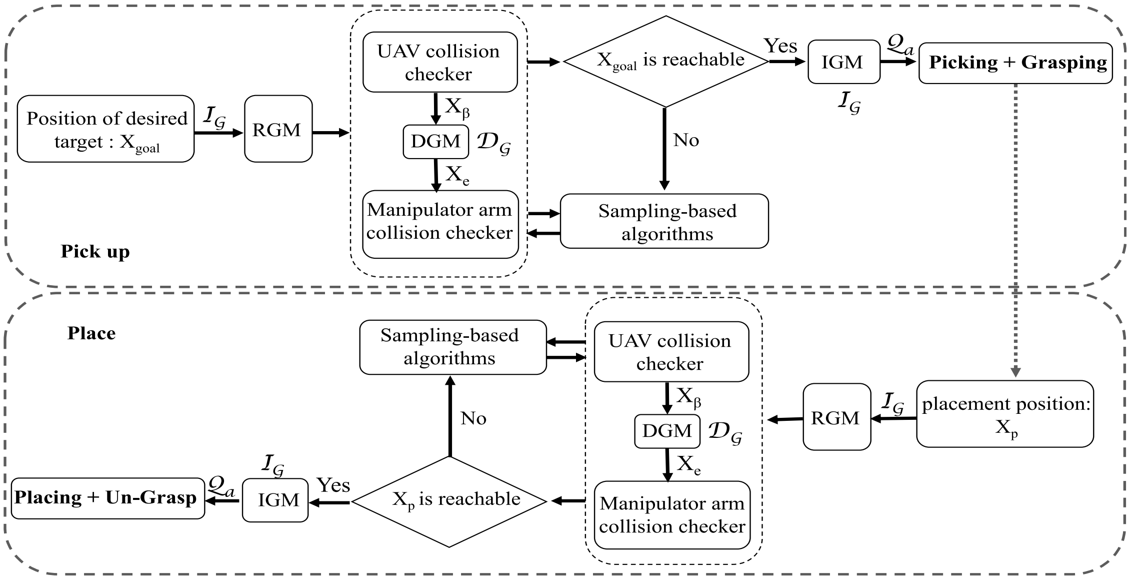


Fig. 10. The path planning phases of the aerial manipulator that must pass to pick and place an object.

The path planning step stands as the initial and vital phase in the aerial pick-and-place process. In our study, we introduce a novel strategy that builds upon both sampling techniques and the geometric model outlined in Sec. 4. This strategy aims to enhance the efficiency and precision of aerial pick-and-place operations.

Figure 10 illustrates a streamlined rendition of the proposed path planning methodology. The process commences with the object of interest situated at a designated position, denoted as X_{goal}. Starting from this X_{goal} location and employing the IGM function $\mathcal{I}_g^+$, the RGM initiates the generation of random points in the vicinity of the target object. These points serve as target locations for the UAV while simultaneously ensuring the manipulator arm's ability to reach the target object.

To iterate this process, a stepping method is employed to generate a new configuration, $X_{\mathcal{B},\text{new}}$, by taking incremental steps in the direction of $X_{\mathcal{B},\text{RGM}}$.

This is done while ensuring that both the UAV and its manipulator arm comply with the DGMs ($\mathcal{D}g^+$) function. Sampling continues until the object becomes reachable, denoted by $X_{\mathcal{B},\text{new}} = X_{\mathcal{B},\text{RGM}}$, signifying that the UAV is prepared to pick up the object.

At this juncture, the algorithm employs the IGM and the $\mathcal{I}g^-$ function to compute the manipulator's articulation vector, denoted as $\mathcal{Q}a$. This vector facilitates the movement of the manipulator's end effector to the X_{goal} position. As the manipulator approaches the pick-up location, the gripper initiates the grasping process. Once the destination for placing the object, X_{p}, is determined, the vehicle transitions to a relative pre-placement and placement position, releasing the target object precisely at the intended destination.

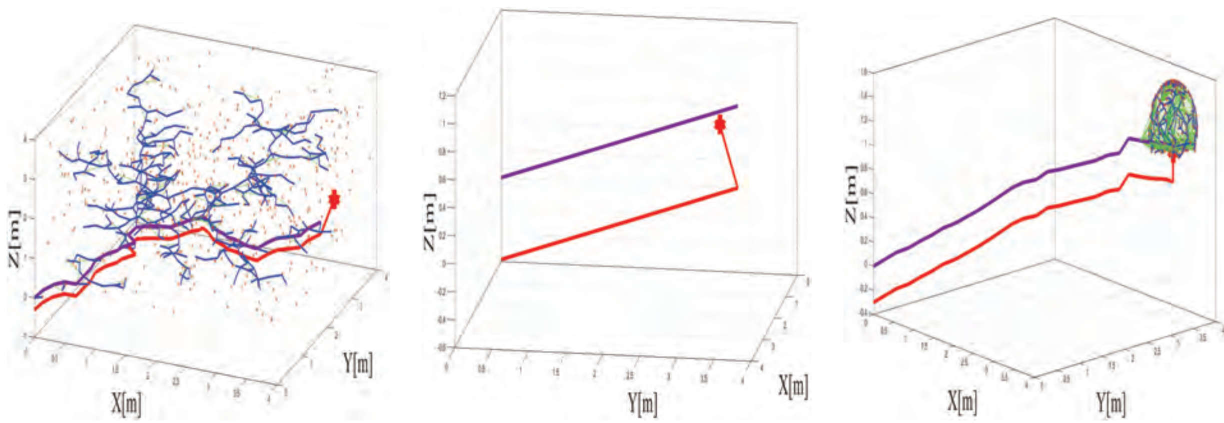
In the following section, we present and dissect the results obtained from our rigorous simulations, providing valuable insights into the performance of the overhead pick-and-place strategies discussed. These results highlight the effectiveness and efficiency of our proposed approach in real-life scenarios, facilitating a comprehensive understanding of its impact and potential applications.

5. Simulation Results and Discussion

To evaluate the efficacy of the proposed S-RGM* algorithm in path planning, we employed MATLAB as the simulation platform for this study. The simulation experiments were conducted in a three-dimensional (3D) environment, initially without any obstacles in the first scenario, and subsequently compared to RRT* and IGM results. In the second scenario, we examined a 2D 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 the systems ($\mathcal{H}$-RRR).

5.0.1. *The first scenario*

In the first scenario, the algorithm generates a trajectory for a UAM with no obstacles in a 3D environment that has dimensions of $4m \times 4m \times 4m$. The desired object being handled by the UAM is located at the position (3.5, 3.5, 1) within the 3D environment. The results obtained by the three approaches S-RGM*, RRT*, and IGM are presented in Figs. 11(a)–11(c) from different perspectives.



(a) (b) (c)

Fig. 11. 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.

Upon completing 500 iterations, as depicted in Fig. 13(c), the path generated by the S-RGM* algorithm quickly converged to the region where the UAM could effectively manipulate the desired object, surpassing the conventional RRT* method Fig. 13(a). This superiority can be attributed to the advantage of using point clouds, which allowed the UAM to perform its task with greater efficiency. These point clouds were generated by the RGM and employed to guide the sampling method towards converging on the ideal position, thereby shortening the path distance between the initial and final points.

Table 2 illustrates a comprehensive performance comparison among three approaches, namely RRT*, IGM, and S-RGM*, in the first scenario. It provides a comparative analysis of the path length for each approach with respect to the UAV and ARM, along with the presence or absence of collisions.

According to the table, the RRT* approach had a path length of 8.6850 for the UAV and 9.1033 for the ARM. Importantly, this method was able to produce a collision-free outcome.

On the other hand, the IGM approach showed a path length of 5.181 for the UAV and 6.0494 for the ARM, but unfortunately, it resulted in a collision.

Table 2. Comparison performance table of three approaches RRT*, IGM, and S-RGM* in the first scenario.

	Path length [UAV]	Path length [ARM]	Collisionfree
RRT*	8.6850	9.1033	Yes
IGM	5.181	6.0494	No
S-RGM*	4.9705	5.2205	Yes

Finally, the S-RGM* approach demonstrated the shortest path length, with 4.9705 for the UAV and 5.2205 for the ARM, and also achieved a collision-free outcome, similar to the RRT* approach.

In conclusion, the S-RGM* approach stands out as the most efficient method in terms of path length for both the UAV and ARM, while ensuring a collision-free outcome.

5.0.2. *The second scenario*

In the second scenario, the algorithm is deployed within an environment where obstacles impede the trajectory towards the desired object. To replicate this scenario, three obstacles are introduced into the 3D environment, as depicted in Fig. 12. Within this scenario, two distinct cases are distinguished for the UAV to attain its target:

(1) In the first case, the aerial manipulator operates within a set and unchanging plane of flight, typically at a constant altitude (Fixed Flight Plane). 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 Fig. 13.
(2) In the second case, the aerial manipulator operates in a more complex, 3D environment, possibly involving changes in altitude, orientation, or other flight parameters (Dynamic Flight Plane). The algorithm operates for 2000 iterations, and the corresponding trees produced by the algorithm are depicted in Fig. 14.

By differentiating between these cases, the algorithm aims to ascertain the most efficient path for the UAM, taking

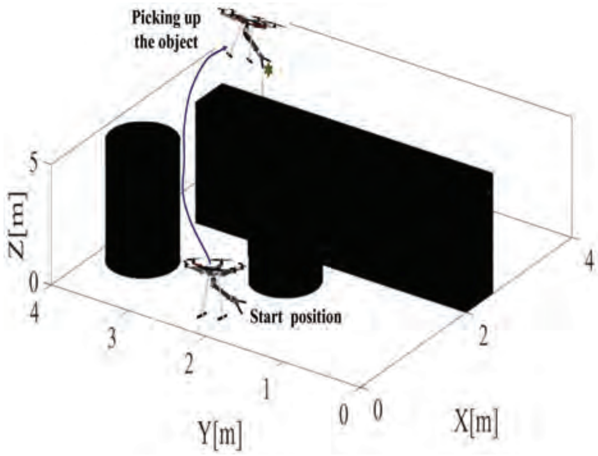
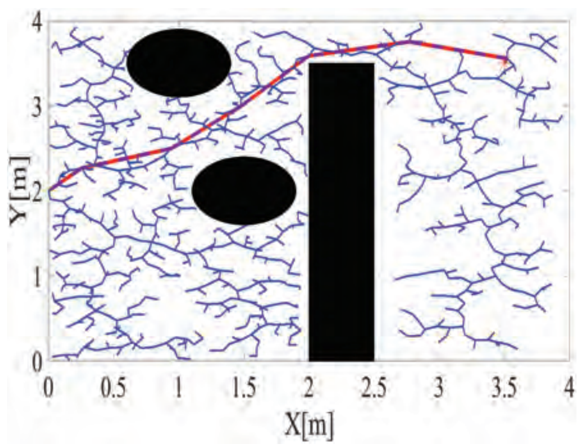


Fig. 12. Aerial pick-up strategy in the second scenario.

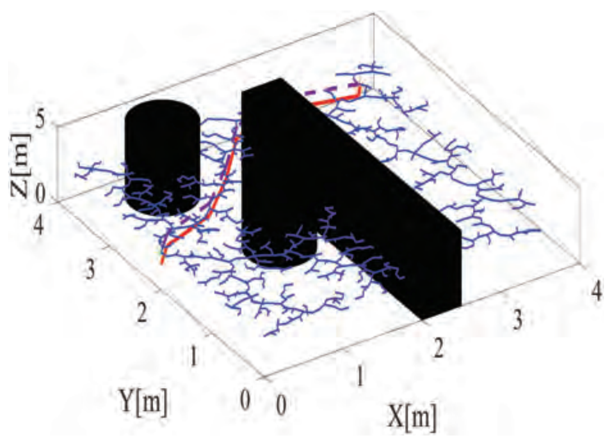
into account both the horizontal and vertical aspects of the environment.

Tables 3 and 4 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.

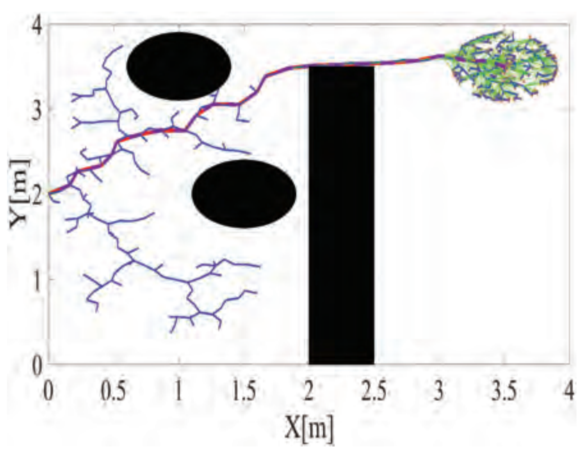
In the first case, the RRT* approach exhibited a sampling time of 82.24 s, using a total of 972 samples. It resulted in a UAV path length of 4.543 m and an ARM path length of 4.94 m. Conversely, the RRT*-RGM* approach achieved a slightly reduced sampling time of 57.78 s with 547 samples, producing a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



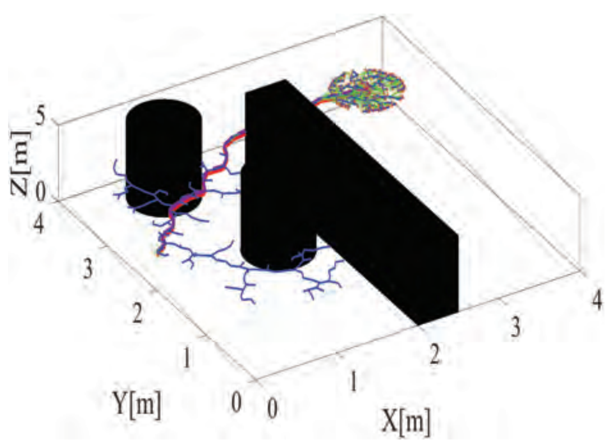
(a) RRT*-2D



(b) RRT*-3D

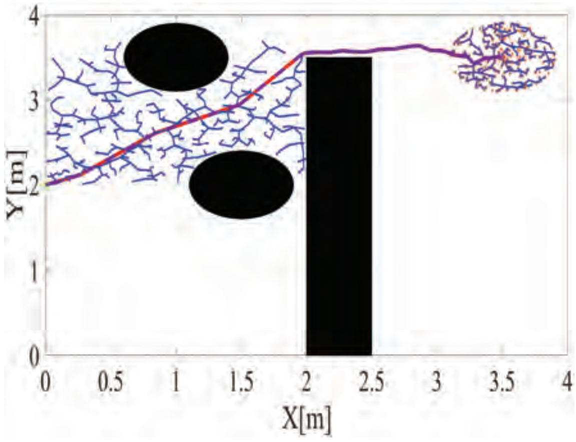


(c) RRT*-RGM*-2D

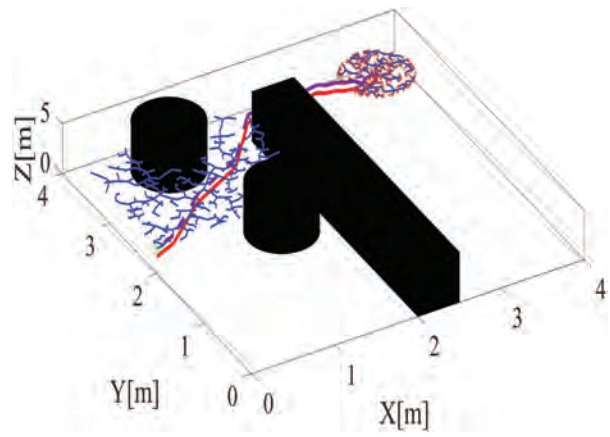


(d) RRT*-RGM*-3D

Fig. 13. 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.

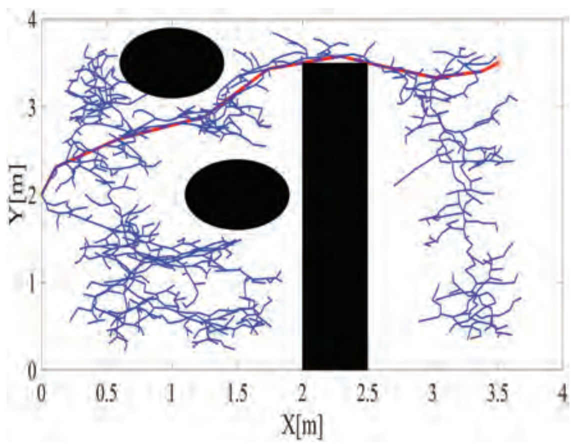


(e) S-RGM*-2D

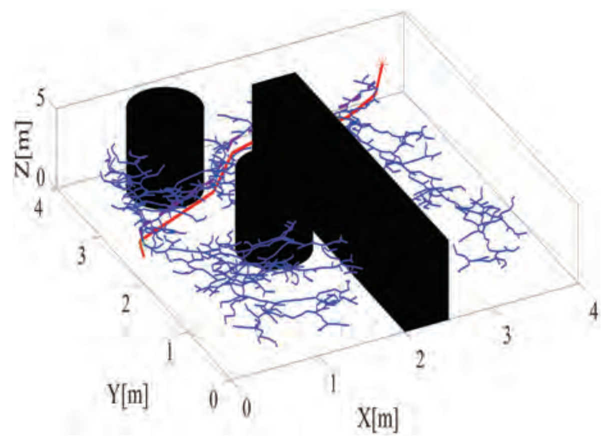


(f) S-RGM*-3D

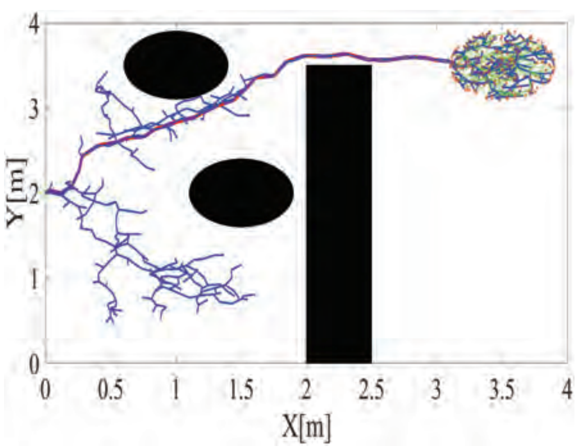
Fig. 13. (*Continued*)



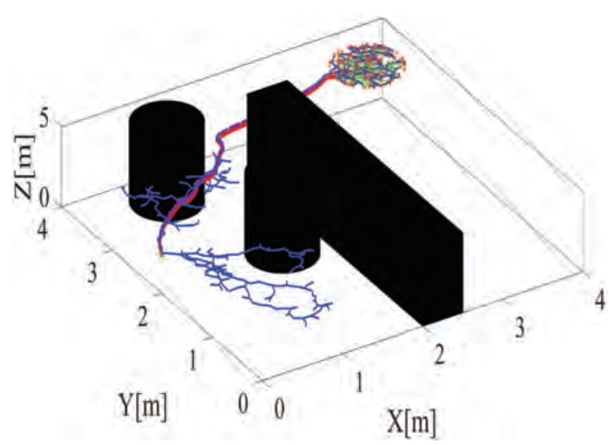
(a) RRT*-2D



(b) RRT*-3D

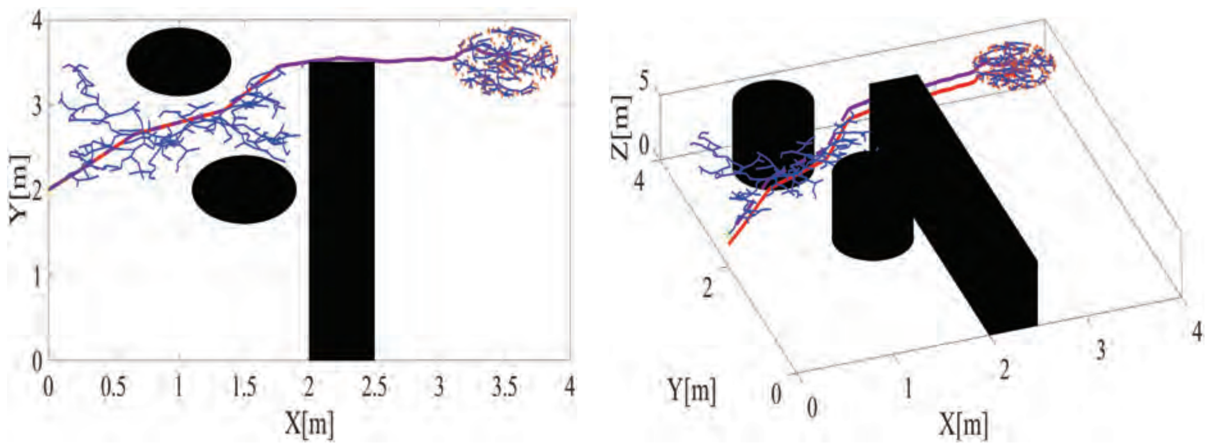


(c) RRT*-RGM*-2D



(d) RRT*-RGM*-3D

Fig. 14. 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.



(e) S-RGM*-2D

(f) S-RGM*-3D

Fig. 14. (*Continued*)

Table 3. Comparison performance table of three approaches RRT*, RRT*-RGM*, and S-RGM* in the second scenario.

Scenario 2	Approach	Sampling time [S]	Sampling number	Path length [UAV] (m)	Path length [ARM] (m)
1	RRT*	82.2411588	972	4.5432720327	4.9454903938
	RRT*-RGM*	57.7834467	547	4.2019639913	4.9374771523
	S-RGM*	34.2245274	475	4.0003855859	4.8003868179
2	RRT*	144.574231	1312	5.7248080639	6.1270264249
	RRT*-RGM*	66.4797705	626	4.5432720327	4.9454903938
	S-RGM*	38.3879616	594	4.3307634876	5.0809982024

Table 4. Comparison performance table of three approaches RRT*, RRT*-RGM*, and S-RGM* in the second scenario.

Scenario 3	Approach	Sampling time [S]	Sampling number	Path length [UAV] (m)	Path length [ARM] (m)
1	RRT*	235.7566	1609	5.9980	6.4002
	S-RGM*	68.3118	577	3.9137	4.4226
2	RRT*	215.3571	1141	6.0232	6.4254
	S-RGM*	64.0025	355	5.8283	6.2010

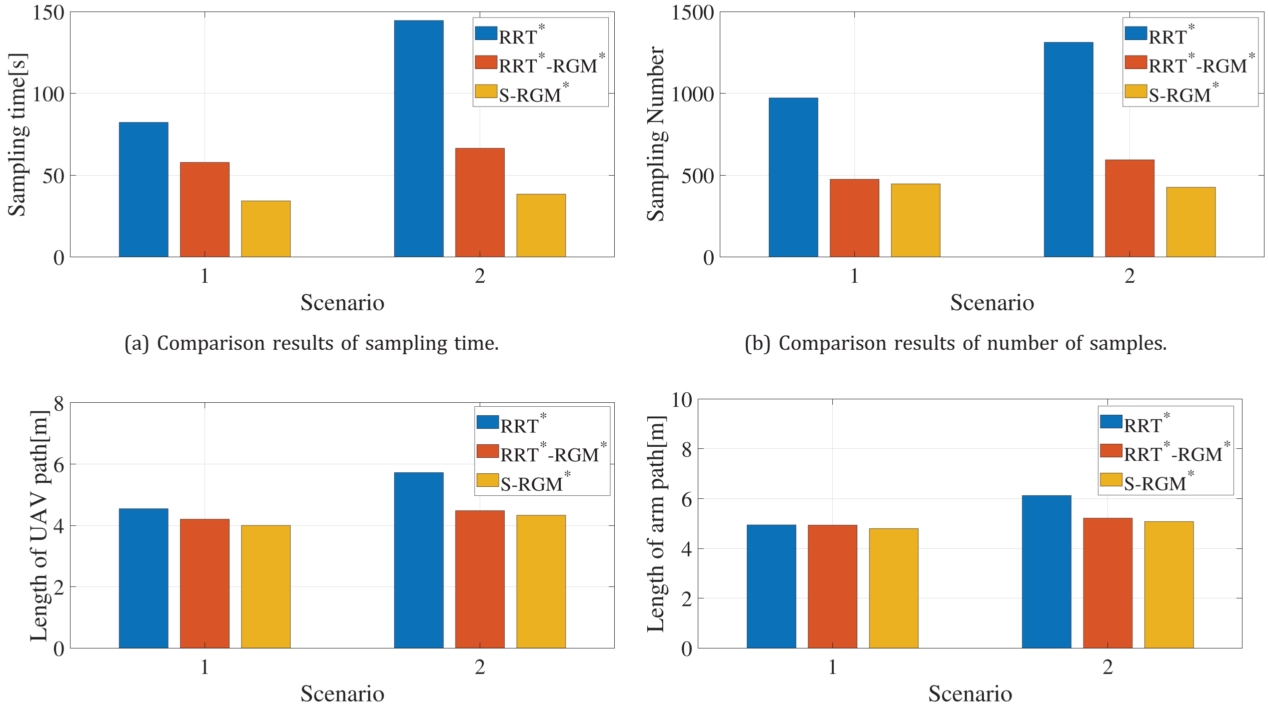
475 samples, leading to a UAV path length of 4.00 m and an ARM path length of 4.80 m.

For scenario 2, the RRT* approach recorded a higher sampling time of 144.57 s, employing 1312 samples. This resulted in a UAV path length of 5.72 m and an ARM path length of 6.12 m. Conversely, the RRT*-RGM* approach demonstrated increased efficiency with a reduced sampling time of 66.47 s and 626 samples. The resulting UAV path length was 4.54 m, while the ARM path length measured 4.94 m. Similarly, the S-RGM* approach showcased notable efficiency with a sampling time of 38.38 s, utilizing 594 samples, resulting in a UAV path length of 4.3307634876 m and an ARM path length of 5.08 m.

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 shown in Fig. 15(a) and sampling number (given in Fig. 15(b). Meanwhile, the RRT* approach showcased higher path lengths for both the UAV and the ARM in scenario 2 (see Figs. 15(c) and 15(d).

5.0.3. *The third scenario*

In this scenario, where an unmanned aerial manipulator has to navigate in a constrained environment with tight spaces,



(a) Comparison results of sampling time.

(b) Comparison results of number of samples.

(c) Comparison results of length of path.

(d) Comparison results of length of path.

Fig. 15. Performance table of three approaches RRT*, RRT*-RGM*, and S-RGM*in the second scenario using bar charts.

the challenge is to find a collision-free path while taking into account the additional complexity of the UAM, such as its manipulator arm. Let's imagine a situation where the UAM has to move through a congested warehouse or dense urban area. There may be narrow gaps between obstacles where the main body of the UAM cannot pass, but the smaller and more flexible UAV base is able to move.

To evaluate the performance of the proposed approach, two separate experiments were carried out comparing the proposed method with RRT*.

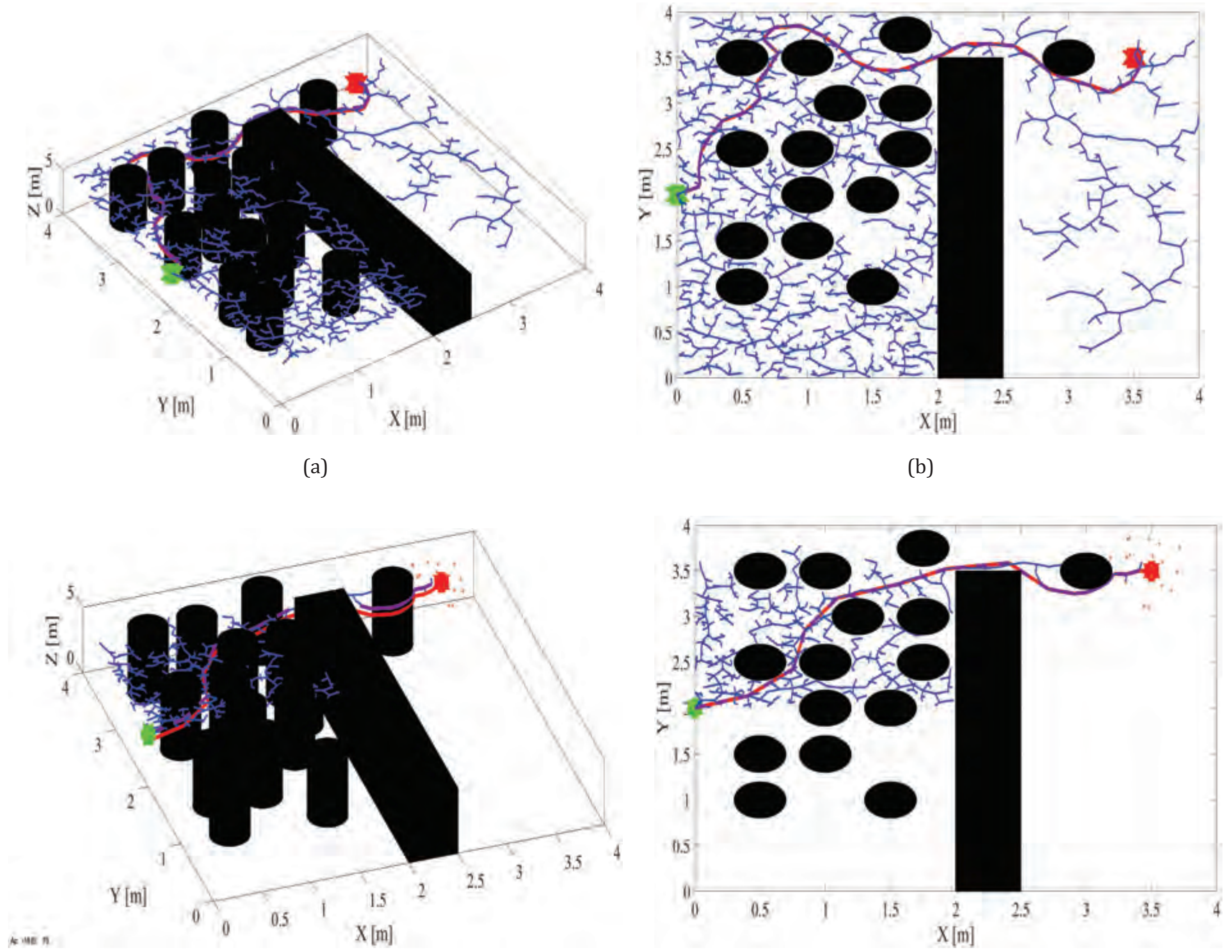
In the first case, the aerial manipulator operates within a set and unchanging plane of flight, typically at a constant altitude (Fixed Flight Plane). Subsequently, the algorithm searches for the optimal path while maintaining a constant altitude. The algorithm undergoes 1500 iterations, and the resulting trees generated by the algorithm are illustrated in Fig. 16.

In the second case, the aerial manipulator operates in a more complex, 3D environment, possibly involving changes in altitude, orientation, or other flight parameters (Dynamic Flight Plane). The algorithm operates for 2000 iterations, and the corresponding trees produced by the algorithm are depicted in Fig. 17.

The table compares the performance of tow path planning algorithms: RRT* and S-RGM*-across several metrics in a specific scenario. The metrics measured include sampling time, the number of samples taken, and the path length for both the UAV base and its manipulator arm. The scenario involves navigating a UAM through an environment, and the goal is to evaluate the computational efficiency and path quality of each method.

Sampling Time: One of the most noticeable differences is in the sampling time required by each method. In both test cases, RRT* requires significantly more time to find a path. In the first test case, RRT takes 235.76 s, whereas S-RGM* only needs 68.31 s. Similarly, in the second case, RRT* takes 215.36 s, while S-RGM* completes the task in 64 s. This demonstrates that S-RGM* is substantially faster, offering 3–4 times more computational efficiency than RRT*. The reduction in sampling time can be particularly beneficial in real-time applications where quick decision-making is crucial.

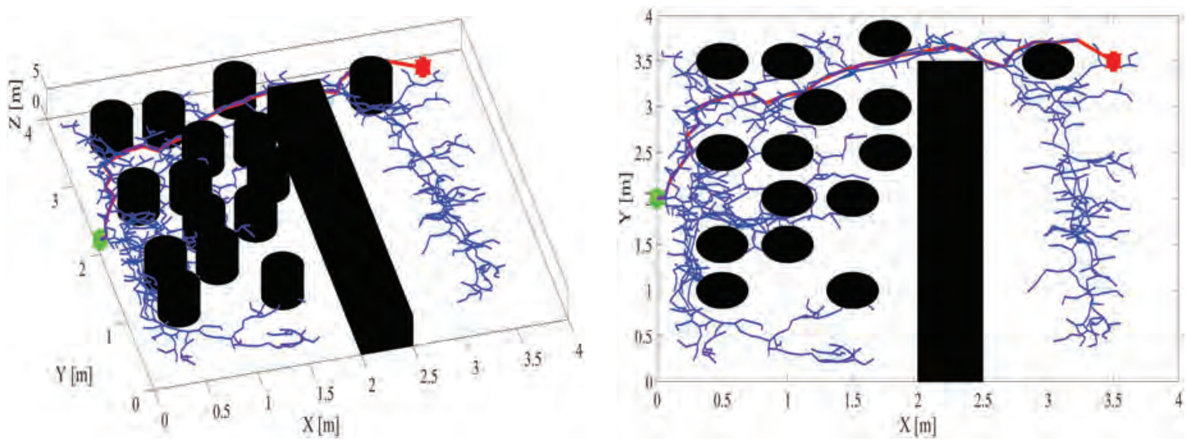
Sampling Number: The number of samples required also differs significantly between the approaches. In the first test case, RRT* generates 1609 samples, while S-RGM* only requires 577 samples to find a path. In the second case, RRT* uses 1141 samples, while S-RGM* generates just 355 samples. The fewer samples required by S-RGM* indicate that it is more efficient in exploring the configuration space, reducing the computational load and time required for path planning.



(a) (b)

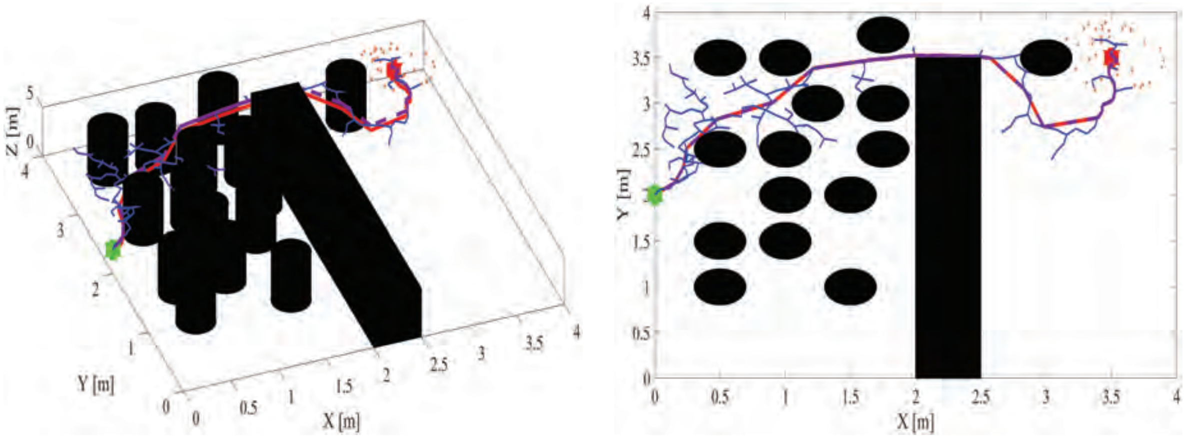
(c) (d)

Fig. 16. The first case of path planning in the third scenario to pick up the object. Each purple and red line shows the trajectory of the aerial manipulator's body and effector position, respectively.



(a) (b)

Fig. 17. The second case of path planning in the third scenario to pick up the object. Each purple and red line shows the trajectory of the aerial manipulator's body and effector position, respectively.



(c)

(d)

Fig. 17. (*Continued*)

Path Length (UAV and ARM): In terms of the path length for both the UAV and the robotic arm, S-RGM* generally provides shorter or comparable paths, which translates into more efficient movement. In the first case, S-RGM* generates a UAV path of 3.91 m and an ARM path of 4.42 m, compared to 5.99 m and 6.40 m for RRT*, respectively. This indicates that S-RGM* finds shorter, more direct paths. In the second case, the path lengths for the UAV and ARM are slightly closer between the two methods, with S-RGM* providing a 5.83 m UAV path and a 6.20 m ARM path, while RRT* produces paths of 6.02 m for the UAV and 6.43 m for the ARM. Even in this scenario, S-RGM* offers marginally more efficient paths.

The results clearly show that S-RGM* outperforms RRT* in both computational efficiency and path quality. It consistently requires less sampling time and fewer samples while generating shorter paths for both the UAV and the manipulator arm. This makes S-RGM* a more suitable choice for environments where quick and efficient navigation is critical, especially in complex and constrained spaces.

5.0.4. *The fourth scenario*

The objective of this scenario is to perform a pick-and-place operation with the aim of transporting an object and precisely depositing it at a predetermined target location. In this task, the aerial manipulator initially takes flight to approach and retrieve the object.

Subsequently, it transports the object to the specified target location and carefully positions it there. A comprehensive overview of this entire experimental process is provided in Fig. 18.

To assess the performance of the proposed approach, two distinct experiments have been conducted:

(1) In the first experiment, the aerial manipulator is configured to maintain a constant altitude while navigating within a fixed plane in the experimental field. The object's position and orientation angles are consistently set at [2.3, 4, 2.5] and 0 degrees, respectively. Likewise, the target location remains stationary at [0.5, 0.5, 2.5].
(2) In the second experiment, the aerial manipulator operates in a dynamic 3D environment. It represents a more challenging and dynamic operational setting where adaptability and obstacle avoidance become crucial. Here, the object's position and orientation angles are maintained at [2.3, 4, 2.5] and 0 degrees, while the target location retains its position at [0.5, 0.5, 0].

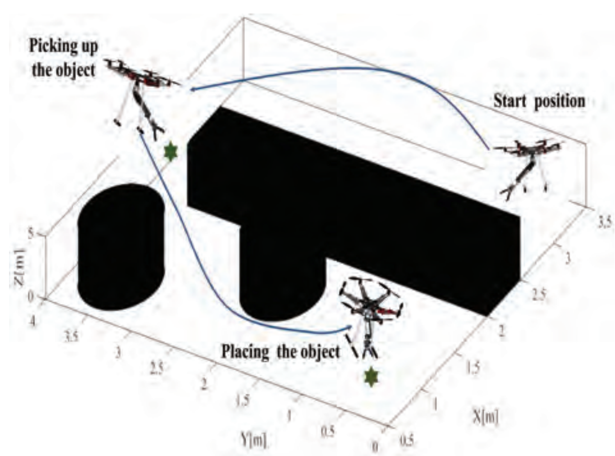


Fig. 18. The aerial pick-up and place strategy in the third scenario.

Table 5. Performance of the aerial pick-up and place strategy using S-RGM* in the third scenario.

Scenario 3	1	2
Sampling time [S]	11.7526	15.1733
Sampling number	181	308
Path length [UAV] (m)	7.0956	7.6089
Path length [ARM] (m)	10.9335	11.8253

As depicted in Fig. 18, there exists an obstacle in close proximity to the object and an additional three obstacles near the designated target placement. The aerial manipulator must navigate carefully to avoid any collisions with these obstacles during the execution of its tasks.

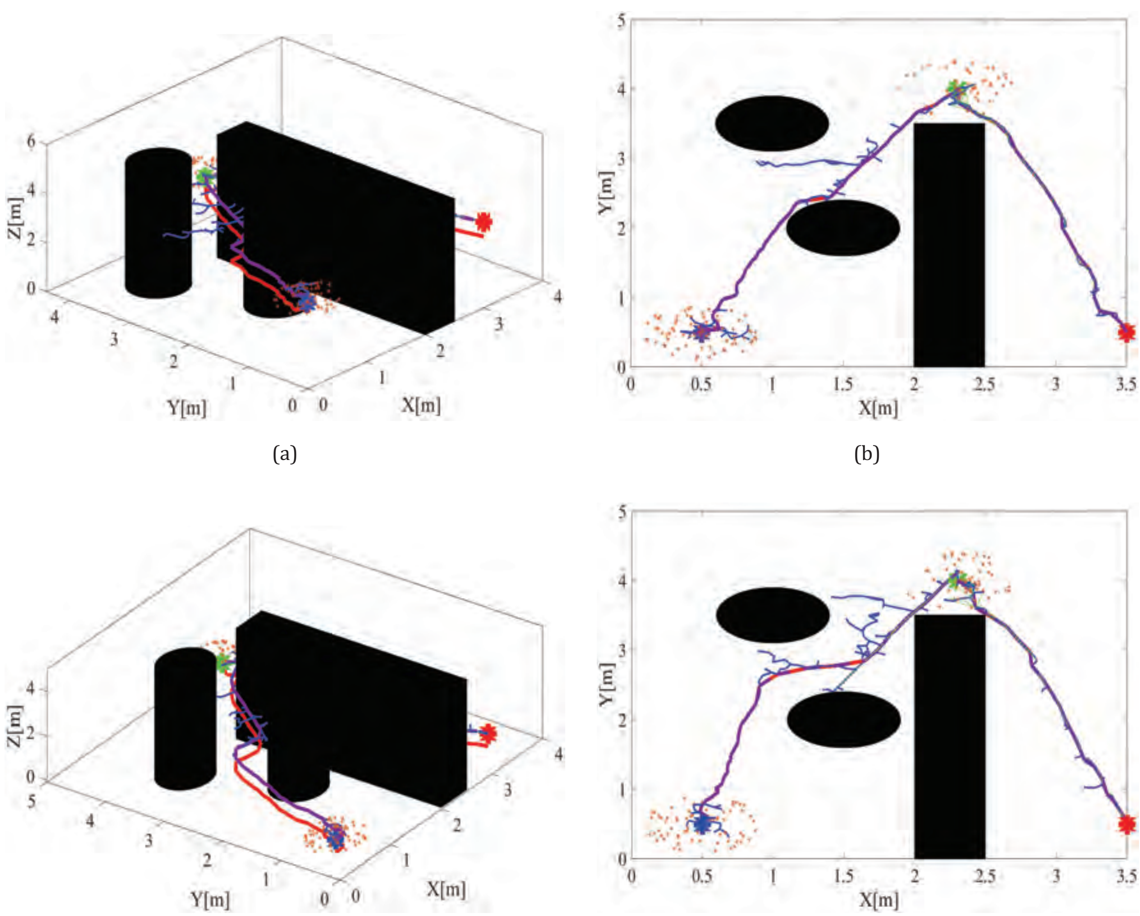
Figures 19(a)–19(d) show the results of scenario 3 for the two aerial pick-and-place experiments. Table 5 presents the performance metrics of the aerial pick-up and place strategy employing the S-RGM* approach in the third scenario. The data provided in this table encapsulate critical aspects of the strategy's execution, offering valuable insights into its efficiency and effectiveness.

In Scenario 3, we observe two distinct experiments, labeled as '1' and '2'. For both experiments, the S-RGM* approach is utilized.

Sampling Time (S): In the first experiment (Scenario 3, Experiment 1), the sampling time is recorded as 11.7526 s. In the second experiment (Scenario 3, Experiment 2), the sampling time slightly increases to 15.1733 s. This metric reflects the time required for the algorithm to generate potential path points and configurations.

Sampling Number: Experiment 1 involves 181 samples, while Experiment 2 utilizes 308 samples. This metric signifies the number of sample points considered during the path planning process, with Experiment 2 requiring a higher number of samples.

Path Length (UAV): The path length covered by the UAV during Experiment 1 is 7.0956 m, whereas in Experiment 2,



(a) (b)

(c) (d)

Fig. 19. Simulation results of the proposed aerial pick-and-place planner under different conditions. The purple line indicates the trajectory of the UAV body position. The red line indicates the trajectories of the link end.

it extends to 7.6089 m. This metric measures the distance traveled by the UAV during the execution of the task.

Path Length (ARM): Experiment 1 results in a path length of 10.9335 m for the manipulator arm, while Experiment 2 reports a path length of 11.8253 m. This metric quantifies the distance traversed by the manipulator arm while manipulating the object.

The table provides a comprehensive overview of the performance of the S-RGM* approach in both experiments, aiding in the assessment of its suitability for the specified aerial pick-up and place scenario.

6. Conclusions

This paper has introduced and thoroughly explored the S-RGM* approach, a novel sampling-based strategy for path planning in the context of aerial pick-and-place operations, incorporating geometric methodologies. Through comprehensive experimentation and analysis, we have demonstrated the efficacy and versatility of this approach across varying scenarios.

Our findings highlight the remarkable performance of S-RGM*, showcasing its ability to generate efficient and collision-free trajectories for both the UAV and its manipulator arm. This approach exhibits adaptability in different environmental conditions, including fixed and dynamic flight planes, while successfully avoiding obstacles. Moreover, the relatively low computational demands of S-RGM* make it a promising choice for real-world applications where timely decision-making and resource efficiency are paramount.

The S-RGM* approach not only streamlines the path planning process but also contributes to the advancement of aerial manipulation systems, enabling them to tackle complex tasks with precision and agility. By seamlessly integrating geometric models and sampling techniques, we have taken a significant step forward in the domain of aerial pick-and-place operations.

In conclusion, the S-RGM* approach exhibits promise as a valuable tool in the repertoire of aerial robotics, opening doors to a wide array of applications that demand meticulous path planning, enhanced safety, and efficiency. Future research may explore further refinements and potential extensions of this approach, ultimately pushing the boundaries of what is achievable in aerial manipulation and beyond.

Acknowledgment

Special thanks go to “Ecole Militaire Polytechnique” for providing the resources necessary for carrying out this research.

ORCID

Zamoum Housseyn https://orcid.org/0000-0002-3971-7426
Bouzid Yasser https://orcid.org/0000-0002-8400-9912
Guiatni Mohamed https://orcid.org/0000-0002-5899-6862

References

[1] A. Boeuf, J. Cortés, R. Alami and T. Siméon, Enhancing sampling-based kinodynamic motion planning for quadrotors, in *2015 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 2447–2452.

[2] A. Bouras, Y. Bouzid and M. Guiatni, Multi-UAV coverage path planning for gas distribution map applications, *Unmanned Syst.* **10**(3) (2022) 289–306.

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

[4] M. Brunner, G. Rizzi, M. Studiger, R. Siegwart and M. Tognon, A planning-and-control framework for aerial manipulation of articulated objects, *IEEE Robot. Autom. Lett.* **7**(4) (2022) 10689–10696.

[5] A. Caballero, M. Bejar, A. Rodriguez-Castano and A. Ollero, Motion planning with dynamics awareness for long reach manipulation in aerial robotic systems with two arms, *Int. J. Adv. Robot. Syst.* **15**(3) (2018) 1729881418770525.

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

[7] C. A. Dimmig, A. Goodridge, G. Baraban, P. Zhu, J. Bhowmick and M. Kobilarov, A small form factor aerial research vehicle for pick-and-place tasks with onboard real-time object detection and visual odometry, arXiv:2308.01398.

[8] G. Dong, F. Yang, K.-L. Tsui and C. Zou, Active balancing of lithium-ion batteries using graph theory and a-star search algorithm, *IEEE Trans. Indust. Inform.* **17**(4) (2020) 2587–2599.

[9] A. Eele and A. Richards, Path-planning with avoidance using nonlinear branch-and-bound optimization, *J. Guidance Control Dyn.* **32**(2) (2009) 384–394.

[10] J. D. Gammell, T. D. Barfoot and S. S. Srinivasa, Batch informed trees (bit*): Informed asymptotically optimal anytime search, *Int. J. Robot. Res.* **39**(5) (2020) 543–567.

[11] J. D. Gammell, S. S. Srinivasa and T. D. Barfoot, Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic, in *2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (IEEE, 2014), pp. 2997–3004.

[12] Z. Housseyn, B. Yasser and G. Mohamed, Modeling and trajectory tracking by robust control of a hexacopter with a 3-d of manipulator arm, in *2022 2nd Int. Conf. Advanced Electrical Engineering (ICAEE)* (IEEE, 2022), pp. 1–7.

[13] L. Janson, E. Schmerling, A. Clark and M. Pavone, Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions, *Int. J. Robot. Res.* **34**(7) (2015) 883–921.

[14] A. E. Jiménez-Cano, D. Sanalitro, M. Tognon, A. Franchi and J. Cortés, Precise cable-suspended pick-and-place with an aerial multi-robot system: A proof of concept for novel robotics-based construction techniques, *J. Intell. Robot. Syst.* **105**(3) (2022) 68.

[15] S. Karaman and E. Frazzoli, Sampling-based algorithms for optimal motion planning, *Int. J. Robot. Res.* **30**(7) (2011) 846–894.

[16] H. Kim, H. Seo, J. Kim and H. Jin Kim, Sampling-based motion planning for aerial pick-and-place, in *2019 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2019), pp. 7402–7408.

[17] H. Lee, H. Kim and H. Jin Kim, Path planning and control of multiple aerial manipulators for a cooperative transportation, in *2015 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 2386–2391.

[18] H. Liu, J. Chen, J. Feng and H. Zhao, An improved RRT* UAV formation path planning algorithm based on goal bias and node rejection strategy, *Unmanned Syst.* **11**(4) (2023) 317–326.

[19] J. K. Medhi and C. L. McGhan, Towards a modular architecture for intelligent aerial manipulator systems, in *2018 AIAA Information Systems-AIAA Infotech@ Aerospace* (2018), pp. 1631.

[20] A. Mohiuddin, T. Tarek, Y. Zweiri and D. Gan, A survey of single and multi-UAV aerial manipulation, *Unmanned Syst.* **8**(2) (2020) 119–147.

[21] A. Ollero, M. Tognon, A. Suarez, D. Lee and A. Franchi, Past, present, and future of aerial robotic manipulators, *IEEE Trans. Robot.* **38**(1) (2021) 626–645.

[22] M. Otte and N. Correll, C-forest: Parallel shortest path planning with superlinear speedup, *IEEE Trans. Robot.* **29**(3) (2013) 798–806.

[23] S. Prakash, Review of recent path planning optimisation algorithms for unmanned combat aerial vehicles, in *AIAA Aviation 2020 Forum* (2020), pp. 2849.

[24] D. Rachmawati and L. Gustin, Analysis of Dijkstras algorithm and a* algorithm in shortest path problem, *J. Phys.: Conf. Ser.* **1566** (2020) 012061.

[25] D. Sun, N. Wan, W. Dai, Y. Zhang and N. Hovakimyan, Control design for an aerial manipulator for pick-and-place tasks, in *AIAA Scitech 2019 Forum* (2019), pp. 1291.

[26] M. Tognon, E. Cataldi, H. A. Tello Chavez, G. Antonelli, J. Cortés and A. Franchi, Control-aware motion planning for task-constrained aerial manipulation, *IEEE Robot. Autom. Lett.* **3**(3) (2018) 2478–2484.

[27] B.-H. W. Khalil and E. Dombre, *Chap. 4 — Inverse Geometric Model of Serial Robots* (2002), pp. 57–84.

[28] M. Yavari, K. Gupta and M. Mehrandezh, Interleaved predictive control and planning for an unmanned aerial manipulator with on-the-fly rapid re-planning in unknown environments, *IEEE Trans. Autom. Sci. Eng.* (2022) 1–16.

[29] G.-Y. Yin, S.-L. Zhou and Q.-P. Wu, An improved RRT algorithm for UAV path planning, *Acta Electon. Sin.* **45**(7) (2017) 1764.

[30] C. Zammit and E.-J. Van Kampen, Comparison between a* and RRT algorithms for 3d UAV path planning, *Unmanned Syst.* **10**(2) (2022) 129–146.

[31] H. Zamoum, Y. Bouzid and M. Guiatni, Robust control and recursive modelling approach for hexarotor manipulator, *Proc. Inst. Mech. Eng., Part I: J. Syst. Control Eng.* **237**(8) (2023) 09596518231156814.

Zamoum Housseyn is a doctor in the field of Robotics and Automation at "Ecole Militaire Polytechnique". He received his Ph.D. in Robotics Engineering from Ecole Militaire Polytechnique. His research interests include UAV manipulation, dynamic modeling, trajectory planning, and control systems. Dr. Zamoum has authored several research papers in leading journals and conferences in robotics and aerial systems.

Bouzid Yasser is a researcher specializing in Control Systems and Robotic Manipulators. He is affiliated with Ecole Militaire Polytechnique. His work focuses on the integration of aerial platforms with robotic manipulators, robust control strategies, and path planning algorithms.

Guiatni Mohamed is a Professor of Robotics at Ecole Militaire Polytechnique. He holds extensive expertise in UAV systems, multi-body dynamics, and advanced control strategies. His contributions to the field include innovative solutions for UAV manipulator integration and applications in complex environments. Prof. Guiatni has supervised numerous research projects and is actively involved in advancing autonomous robotic systems.