

Unified Robust Path Planning and Optimal Trajectory Generation for Efficient 3D Area Coverage of Quadrotor UAVs

Fatemeh Rekabi-Bana^{ID}, Junyan Hu^{ID}, Tomáš Krajiník, and Farshad Arvin^{ID}

Abstract—Area coverage is an important problem in robotics applications, which has been widely used in search and rescue, offshore industrial inspection, and smart agriculture. This paper demonstrates a novel unified robust path planning, optimal trajectory generation, and control architecture for a quadrotor coverage mission. To achieve safe navigation in uncertain working environments containing obstacles, the proposed algorithm applies a modified probabilistic roadmap to generating a connected search graph considering the risk of collision with the obstacles. Furthermore, a recursive node and link generation scheme determines a more efficient search graph without extra complexity to reduce the computational burden during the planning procedure. An optimal three-dimensional trajectory generation is then suggested to connect the optimal discrete path generated by the planning algorithm, and the robust control policy is designed based on the cascade NLH_{∞} framework. The integrated framework is capable of compensating for the effects of uncertainties and disturbances while accomplishing the area coverage mission. The feasibility, robustness and performance of the proposed framework are evaluated through Monte Carlo simulations, PX4 Software-In-the-Loop test facility, and real-world experiments.

Index Terms—Area coverage, path planning, probabilistic road map, robust control, optimal trajectory, unmanned aerial vehicles.

I. INTRODUCTION

AREA coverage is an active research topic that has been explored by researchers in recent years for a wide range of robotic problems. Many real-world applications have used such techniques by describing a different mathematical framework to achieve a particular solution [1], e.g., underwater mapping, area exploration, search and rescue, and many hazardous or challenging tasks [2], [3]. Due to the advancement in communication, computing and sensing methods, deployment

Manuscript received 14 October 2022; revised 20 April 2023 and 13 August 2023; accepted 21 September 2023. Date of publication 17 October 2023; date of current version 18 April 2024. This work was supported in part by the EU H2020-FET-OPEN RoboRoyale under Project 964492 and in part by the OP VVV under Grant CZ.02.101/0.0/0.0/16_019/0000765RCI. The Associate Editor for this article was M. Yang. (Corresponding authors: Junyan Hu; Farshad Arvin.)

Fatemeh Rekabi-Bana, Junyan Hu, and Farshad Arvin are with the Swarm and Computational Intelligence Laboratory (SwaCIL), Department of Computer Science, Durham University, DH1 3LE Durham, U.K. (e-mail: fatemeh.rekabi-bana@durham.ac.uk; junyan.hu@durham.ac.uk; farshad.arvin@durham.ac.uk).

Tomas Krajiník is with the Artificial Intelligence Center, Czech Technical University in Prague, 166 36 Prague, Czechia (e-mail: tomas.krajnik@fel.cvut.cz).

Digital Object Identifier 10.1109/TITS.2023.3320049

of unmanned aerial vehicles (UAVs) for uncertain area coverage has attracted increasing attention from both academia and industry, which motivates the need to develop more efficient and robust planning and control solutions.

The solution for an autonomous area coverage problem includes path planning to determine a safe route and a control policy to guarantee a precise trajectory following. Accordingly, the design criteria considered in the problem formulation affect the coverage performance significantly. Many researchers try to design path-planning algorithms using various approaches, from classical Road Map frameworks to complex optimisation-based methods to make architectures for using different criteria regarding the robot and workspace conditions [4]. For instance, Galceran and Carreras [1] made a classification and categorised the coverage planning approaches based on their attributes for the area partitioning. This classification demonstrates an explicit perspective of the different approaches regarding environmental data utilisation. Another review suggested a classification based on the optimisation technique used in various algorithms to solve the coverage problem to establish the important cost function contribution to the coverage performance and the main achievements of the problem [5]. Furthermore, different optimisation techniques influence the final results considerably. For instance, [6] suggested a random walk method for heuristic search to improve the coverage efficiency for a static target point. In [7], both Dynamic programming and heuristic optimisation were applied to reduce the travelling cost for a UAV conducting a coverage mission. In another approach, a sampling-based method was proposed to analyse and improve the probability of coverage completeness and convergence for a specific structural inspection [8]. Other approaches exploited chaotic model [9], artificial potential field [10], Spanning Tree Covering [11], cellular and polygon decomposition [12], Probabilistic road map [13], particle swarm [14], neural network [15], and reinforcement learning [16] to optimise different criteria for travelling time, energy consumption, trajectory smoothness, and collision risk.

Although these algorithms considered an established optimisation framework to find a solution for the planning problem, the quadrotor's closed-loop behaviour was not considered in the design criteria, especially in the presence of external disturbances or another source of uncertainties. Such planning solutions may result in trajectories that are not consistent with

the system's dynamic, actuation, or control policy restrictions. Other approaches tried to use the probabilistic attitude towards the planning problem and add some feedback to increase the robustness against the environmental uncertainties [17]. However, the quadrotor dynamic response to the planning results has not been considered and therefore, the same problem might arise in spite of considering environmental feedback.

Moreover, it is important to consider the planning criteria regarding the controller requirements to guarantee the performance of the quadrotor during the coverage mission in dynamic environments [18], [19]. However, many studies didn't make an interactive combination of a planning scheme and a control strategy to develop an integrated management unit to confront uncertainties and disturbances while it is determining the required trajectory for the coverage mission. It is necessary to make a consistent combination of a planning strategy, an optimisation method for trajectory generation, and the control algorithm to develop a robust and efficient framework for the coverage mission regarding the quadrotor's closed-loop response. Using a cascade architecture allows the development of a joint connected planning and control system [20], [21].

The cascade design approach is one of the efficient solutions that can be applied to fill the gap between the planning and control algorithms consistently [22]. It allows the designers to use the advantages of different methods in a unified algorithm [23]. The cascade design results in a hierarchical structure with modules that have to interact with each other while they can be designed and optimised individually [24]. While the structure of the modules is hierarchical, our approach ensures that information is propagated also from the lower (control) module the higher (planning) one, making the system to go beyond a set of simple hierarchically arranged modules, which optimise the problem in separate.

The probabilistic Road Map (PRM) path planning algorithm [25] is one of the promising approaches used by many researchers to develop various planning schemes for narrow spaces [26], underwater coverage mission [27], dynamic trajectory design [28], and many other applications which establish the exclusive capability of this method to be adjusted with different planning objectives. In addition, some research establishes the adaptability of PRM in conjunction with Linear quadratic Gaussian (LQG) to make a unified control-planning algorithm [29], [30]. PRM algorithm has many advantages over other planning algorithms such as being more efficient, versatile and scalable. Also, it has the potential for robustness if the roadmap generation is designed considering a proper approach to describe the environmental changes [31]. A good comparison between the PRM method and other sampling-based planning algorithms is presented in [32] which makes a clear vision of PRM's advantages over other algorithms. Nevertheless, the basic PRM framework needs to be improved to consider the collision risk and be more robust against environmental uncertainties. Otherwise, the output trajectory doesn't guarantee an acceptable collision risk, especially in the presence of uncertainties.

The primary advantage of risk-aware path planning is that it helps ensure the safety of the robot and the surrounding environment. For instance, [33] and [34] proposed two risk-aware path planning algorithms for ground and aerial robots in partially known environments with uncertainties. Those algorithms were developed based on deterministic path planning methods and improved the basic algorithms' performance by adding the risk-awareness function obtained from apriori data. Furthermore, other research such as [35] and [36] showed the risk-aware form of sampling-based motion planning algorithms can improve the original algorithm's performance in uncertain or dynamic environments. In addition, increasing reliability, efficiency, and adaptability are other motivations that persuade using this feature in the novel path planning algorithms and make a better awareness for the robots working in uncertain areas [37], [38].

The second important part of the cascade architecture is the control algorithm, which insures the robust performance of the closed-loop dynamic in the presence of uncertainties, disturbances, and measurement noise. Although various algorithms developed in recent years to achieve this objective, the nonlinear H_∞ algorithm established a promising performance in 3D trajectory tracking and keeping the system's dynamic states and the outputs bounded [39]. However, such algorithms can guarantee closed-loop stability and robustness under specific circumstances for input trajectories coming from the planning algorithm. Accordingly, making a consistent motion planning algorithm is necessary to avoid unstable behaviour.

This paper proposes a novel integration between a robust nonlinear control algorithm and a new version of PRM planning algorithm to make a consistent mission management framework. In spite of the other state-of-the-art planning technique, the proposed algorithm illustrates a hybrid objective function for optimisation. The objective function considers the controller's performance in conjunction with the travelling distance. Accordingly, the planning result will establish a robust closed-loop behaviour while it will optimise the flight trajectory regarding the travelling distance and time. Therefore, the main novelty of this paper is to make a hybrid optimisation framework to guarantee planning performance and closed-loop robust response simultaneously. The resulting framework prevents the planning algorithm from generating high-risk trajectories regarding the system's dynamic, controller features, environmental uncertainties, and disturbances.

The genetic algorithm is used to solve the planning optimisation problem due to the defined cost function's complexity and nonlinear essence. The Monte-Carlo simulation is used to determine the overall robust performance of the suggested algorithm by statistical analysis. In the Monte-Carlo simulation, the coverage scenario was simulated for 500 different sets of uncertainties for the obstacles' boundaries, exogenous disturbances, measurement noise for position and velocity, and parameters' uncertainties. Furthermore, the PX4 recommended environment for software and hardware in the loop, including the Gazebo simulator physical engine, PX4 flight management unit, and the MAVROS communication protocol, has been considered to examine the algorithm's performance for

practical cases. Furthermore, The algorithm's performance is evaluated by conducting flight experiments using the Crazyflie micro-drone. The results obtained from numerical simulations, SIL tests, and flight experiments demonstrate a promising robust performance in the coverage mission. In conclusion, the main contributions of this paper can be summarised as follows:

- 1) Development of a new unified robust planning algorithm for a quadrotor to accomplish an autonomous area coverage mission efficiently in an arbitrary three-dimensional environment that includes obstacles.
- 2) Modifying the basic PRM algorithm by considering risk assessment functions to mitigate the collision risk in order to establish the planning algorithm's robustness in the presence of environmental uncertainties.
- 3) Applying a cascade architecture for system management. This architecture uses optimisation to solve the planning algorithm considering the system's closed-loop response involving the output-feedback NLH_∞ in order to design a consistent trajectory with system dynamic, actuation, and control policy restrictions.

The rest of this paper is organised as follows: the second section is devoted to presenting some essential preliminaries for the modelling and the problem statement. The third section demonstrates the overall design procedure for the integrated control-planning algorithm. The robust PRM algorithm is illustrated in the fourth section and the output-feedback NLH_∞ control method is explained in the subsequent section. The simulation procedure and results are presented in the sixth section and the final part includes the final interpretation and conclusion.

II. PROBLEM FORMULATION

This section illustrates the problem statement and the main assumptions considered to define the mathematical outlines of the problem. Moreover, The main criteria used to solve the problem are stated in this section. The first part is devoted to the mathematical representation of the environment, the second part defines the dynamic model for the agents, and the third subsection describes the area-coverage mission for a flying quadrotor.

A. Environment Modeling

In this section, the mathematical representation of a three-dimensional environment includes a definite boundary and a limited number of obstacles with uncertain prior information. The mission area is defined as $\mathcal{E} = \mathcal{E}^f \cup \mathcal{E}^o$, where \mathcal{E}^f and \mathcal{E}^o represent the free and occupied space respectively. The free space is described as:

$$\mathcal{E}^f = \{q^f \in \mathbb{R}^3 | q^f \notin \mathcal{E}^o, |q^f(x) - x_0^{\mathcal{E}}| \leq x_m^{\mathcal{E}}, \\ |q^f(y) - y_0^{\mathcal{E}}| \leq y_m^{\mathcal{E}}, |q^f(z) - z_0^{\mathcal{E}}| \leq z_m^{\mathcal{E}}\}, \quad (1)$$

where $[x_0^{\mathcal{E}} \ y_0^{\mathcal{E}} \ z_0^{\mathcal{E}}] \in \mathbb{R}^3$ and $[x_m^{\mathcal{E}} \ y_m^{\mathcal{E}} \ z_m^{\mathcal{E}}] \in \mathbb{R}^3$ are defined to describe the predetermined mission area boundaries and central point. Also the prior information about the obstacles'

boundaries is described as:

$$\mathcal{O}^b = \{\mathcal{O}_k^b\}; k = 1, \dots, n_{obs}, \\ \mathcal{O}_k^b = \{q_k^{b_i} + \delta_k^{b_i}\}; i = 1, \dots, n_k, \quad (2)$$

where $q_k^{b_i}$ is the i th known point on the boundary of the k th obstacle, and $\delta_k^{b_i} = \mathcal{N}(0, \sigma_{\delta_q})$ is a Gaussian distributed uncertainty vector considered for $q_k^{b_i}$.

B. Quadrotor Dynamic

This section presents the dynamic modelling of a flying quadrotor with six degrees of freedom according to [39] and [40]. Accordingly, the dynamic behaviour can be modelled as:

$$m\ddot{\xi} + K_\xi \dot{\xi} + mG = R_t(\eta)\mathcal{F}(t) + d_\xi, \\ J(\eta)\ddot{\eta} + C_m(\eta, \dot{\eta})\dot{\eta} = \mathcal{T} + d_\eta, \quad (3)$$

where, $\xi \in \mathbb{R}^3$ and $\eta \in \mathbb{R}^3$ describe the position and attitude vectors for the quadrotor, respectively, m represents vehicle's mass, $J(\eta) \in \mathbb{R}^{3 \times 3}$ is the inertia matrix determined in the inertial frame, $K_\xi \in \mathbb{R}^{3 \times 3}$ is the aerodynamic drag coefficient matrix and $C_m(\eta, \dot{\eta}) \in \mathbb{R}^{3 \times 3}$ is a matrix determined from the Coriolis, centrifugal, and rotational drag effects together. The $\mathcal{F}(t) \in \mathbb{R}^3$ depicts the force vector in the body-fixed frame and the control torque in the inertial frame is denoted by $\mathcal{T} \in \mathbb{R}^3$. The transformation matrix from the body to the inertial frame is denoted by $R_t(\eta) \in \mathbb{R}^{3 \times 3}$ and the gravity acceleration vector in the inertial frame is represented by $G \in \mathbb{R}^3$. The effects of external disturbances and modelling uncertainties are defined by $d_\xi \in \mathbb{R}^3$ and $d_\eta \in \mathbb{R}^3$ for translational and rotational dynamics. It is also assumed that these terms belong to $\mathcal{L}_2(0, \infty)$.

C. Problem Statement

In this paper, it is assumed that the quadrotor can fly through the free space in the mission area and gather the required data using a set of sensors to collect three-dimensional points in a predetermined area around the quadrotor. According to the sensor package capabilities, the quadrotor should stand in a limited number of target positions to cover the overall mission area. Therefore, the coverage mission can be reduced to a problem that includes the following steps:

- 1) Finding a set of target points in the free space to cover the overall mission area
- 2) Motion planning for the quadrotor to cover all the target points
- 3) Trajectory generation and tracking to control the quadrotor through the environment

The required number of target points depends on the area complexity, the coverage capability of the quadrotor at each point, and the flying time restriction due to limited energy resources for the quadrotor. Accordingly, if $Q^{tg} = \{q_i^{tg}\}$, $i = 1, \dots, N_{tg}$ is considered as a set of target points to cover the overall area, the quadrotor can obtain a reward when it passes

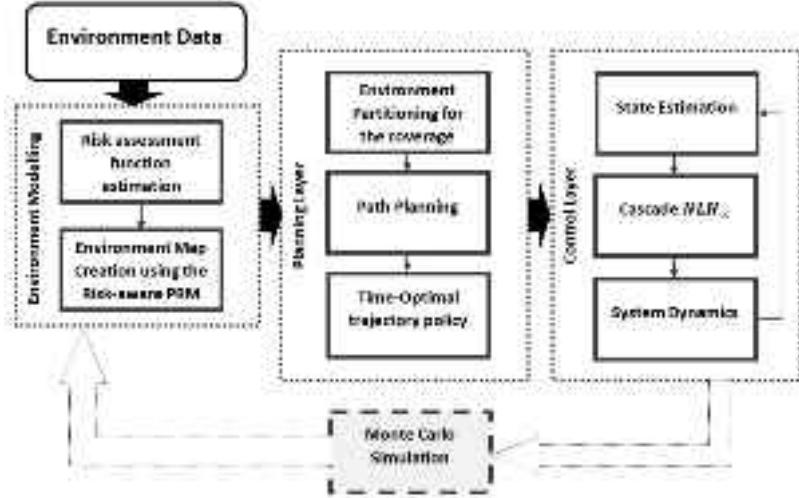


Fig. 1. Integrated Algorithm Flowchart which demonstrates three different layers to generate the environment's map, make a plan for the coverage, and control the drone through the planned trajectory.

through the vicinity of each target points for the first time. The reward function can be described as:

$$\mathcal{R} = \frac{1}{N_{tg}} \sum_{i=1}^{N_{tg}} \mathbf{r}_i,$$

$$\mathbf{r}_i = \begin{cases} \mathbf{r}_{DA} & \text{if } |\xi(t) - q_i^{tg}| \leq d_{arr} \quad t_i^{lv} - t_i^{fav} \geq \delta t_{DA} \\ \mathbf{0} & \text{otherwise} \end{cases}$$
(4)

where, \mathbf{r}_{DA} is the reward value for data acquisition at each target point, d_{arr} is the distance to establish the arrival criteria, t_i^{fav} is the time flag for the quadrotor's first-time arrival at q_i^t , t_i^{lv} is the time flag for the moment that quadrotor leaves the q_i^t target point, and δt_{DA} is the time required for data acquisition at each target point. If Π is considered as a motion plan for the quadrotor to cover the area through target points, then the value function for the mission can be described as:

$$\mathcal{V}_{mission}(\Pi) = (\gamma^t \mathcal{R} - \mathcal{C}_{mission})_{|\Pi}, \quad \gamma \in (0, 1],$$
(5)

where, $\mathcal{C}_{mission}$ is a cost function and depends on the tracking accuracy, required control effort for the mission, and the travelling distance through the area. γ is a coefficient that should be positive and less than 1, and t is the mission accomplishment time. Accordingly, the value function expressed in (5) includes the reward function decreasing by γ^t factor as the mission accomplishment time is increasing. (6) describes the cost function as follows:

$$\mathcal{C}_{mission} = \sum_{i=1}^{N_{tg}} \mathbf{J}_i^C + \int_0^{T_{mis}} \|\mathbf{z}_\xi\|_2^2 d\tau,$$
(6)

where \mathbf{J}_i^C is the travelling distance between each pair of target points in the mission area, N_{tg} is the number of target points considered for the coverage, and \mathbf{z}_ξ is an objective function that depends on the tracking accuracy and the control effort required to maintain the system robust against the uncertainties, disturbances, and noise. Accordingly, the robust

coverage mission can be defined as follows:

$$\begin{aligned} \Pi^* = & \text{argmax} (\sup(\mathcal{V}_{mission})) \\ \text{s.t. } & \mathbf{T}(\Pi) \leq \mathbf{T}_{ad}. \end{aligned}$$
(7)

D. Integrated Control-Planning Algorithm

The integrated control and planning algorithm should be capable of maximizing the coverage value function as described in (5). The cost function is defined in (6) which is a function of tracking accuracy, control effort, and the travelling distance simultaneously. According to previous sections, the suggested robust PRM algorithm finds an optimal solution for the travelling distance and the cascade NLH_∞ guarantees robustness against the uncertainties, disturbances and measurement noise. Also, trajectory planning implies the minimum time constitution meanwhile it establishes the continuity requirements for the desired trajectory according to the control algorithm's requirements. On the other hand, the suggested planning solution depends on the control algorithm's robust performance to maintain the failure risk below the acceptable threshold via the risk-awareness feature. Consequently, the integrated algorithm presented in Fig.1, demonstrates the relation between different layers including three main subroutines to make a consistent mission management framework. Moreover, Figure 1 shows how the planning algorithm is affected by the results obtained from the control algorithm through the Monte Carlo simulation. The relation between those two layers is discussed in detail through section III-A.

III. RISK-AWARE PRM ALGORITHM

In this section a new robust PRM algorithm is proposed to solve the area coverage mission as stated in (7) for a flying quadrotor in an uncertain environment. The suggested algorithm has a structure as described in Algorithm.1:

The following subsections discuss each of the algorithms proposed to complete the overall planning as stated in Alg.1

Algorithm 1 The Proposed Robust PRM

Input: $\mathcal{E}, \mathcal{O}^b$ - space boundaries for exploration and obstacle boundaries prior information
Output: Π^* - coverage plan
 $\mathcal{P}(\mathcal{O}^b) \leftarrow \text{Alg.2 finds risk assessment function}$
 $\mathcal{G}(\mathcal{E}^f, \mathcal{P}(\mathcal{O}^b)) \leftarrow \text{Alg.3 finds search graph}$
 $Q^t \leftarrow \text{Alg.4 finds target points}$
 $\Pi^* \leftarrow \text{Alg.5 finds the coverage plan}$
where $\mathcal{P}(\mathcal{O}^b)$ represents the risk assessment function, $\mathcal{G}(\mathcal{E}^f, \mathcal{P}(\mathcal{O}^b))$ is the graph which will be used for planning, Q^t demonstrates the set of target points for the coverage, and Π^* stands for the mission plan obtained from the proposed algorithm.

A. Environment Mapping

In this section, a new two-stage algorithm is proposed to obtain a fully connected search graph for the mission area, which considers the collision risk with the obstacles based on uncertain prior information. The first stage is devoted to estimating an assessment function to evaluate the collision risk at each point in free space. As stated before, a prior information set includes a few points on the boundary of each obstacle. The suggested algorithm splits the data into two parts: 1) $\mathcal{O}_{x,y}^b$, which is a set of projected points on the horizontal plane, and 2) \mathcal{O}_z^b , which is a set that includes the z components of the points in \mathcal{O}^b . A normalized elliptical function can be associated with each arbitrary set of points placed on a boundary in a plane, and this function can be estimated if there are at least five points in the set. The elliptical function can be stated as follows:

$$\begin{aligned} f^{el}(x, y) &= [\Theta^{el}] [\mathcal{S}_{el}(x, y)]^T + 1, \\ [\Theta^{el}] &= [\theta_1^{el} \theta_2^{el} \theta_3^{el} \theta_4^{el} \theta_5^{el}], \\ \mathcal{S}_{el}(x, y) &= [x^2 \ y^2 \ xy \ x \ y]. \end{aligned} \quad (8)$$

This elliptical function splits the space into three segments based on its value. The function's value varies from $-\infty$ to zero outside the boundary, becomes zero on the boundary, and varies from zero to its maximum value inside the boundary if the boundary encloses the origin, otherwise, the function value becomes positive outside and negative inside the boundary. Therefore, this function could be applied to assess the proximity to the boundary of the obstacle. According to (8), if n_k points exist for the k_{th} obstacle's boundary, the best peripheral elliptic function can be estimated by the following equation:

$$\begin{aligned} [\hat{\Theta}_k^{el}] &= [\mathcal{A}_k]^\dagger [-\mathbf{1}]_{n_k}, \\ [\hat{\Theta}_k^{el}] &= [\hat{\theta}_1^{el} \hat{\theta}_2^{el} \hat{\theta}_3^{el} \hat{\theta}_4^{el} \hat{\theta}_5^{el}], \\ [\mathcal{A}_k] &= \begin{bmatrix} \mathcal{S}_{el}(q_k^{b_1}(x), q_k^{b_1}(y)) \\ \mathcal{S}_{el}(q_k^{b_2}(x), q_k^{b_2}(y)) \\ \vdots \\ \mathcal{S}_{el}(q_k^{b_{n_p}}(x), q_k^{b_{n_p}}(y)) \end{bmatrix}, \\ [-\mathbf{1}]_{n_k} &= [-1 \ -1 \ \dots \ -1]_{n_p \times 1}^T, \end{aligned} \quad (9)$$

where $[\cdot]^\dagger$ means pseudo inverse. The collision risk is directly related to the distance between the quadrotor and the obstacle boundary. Therefore, the estimated elliptical function can be used as a criterion to evaluate the collision risk as presented in Alg.2

Algorithm 2 Risk Assessment Function Evaluation

Data: \mathcal{O}^b - obstacles' boundaries prior information
Result: $\mathcal{P}(\mathcal{O}^b) \leftarrow \{\mathcal{P}_k(\mathcal{O}_k^b)\}_{k=1:n_{obs}}$ - risk assessment function
 $\lambda \leftarrow \text{Monte Carlo Simulation}$
for $k = 1 : n_{obs}$ **do**
 $[\hat{\Theta}_k^{el}] \leftarrow (9)$
if origin is enclosed **then**
 $\mathcal{P}_k(\mathcal{O}_k^b) = -\frac{\lambda}{\hat{f}_k^{el}(x, y)}$
else
 $\mathcal{P}_k(\mathcal{O}_k^b) = \frac{\lambda}{\hat{f}_k^{el}(x, y)}$

where n_{obs} is the number of distinct occupied regions in the workspace.

According to the equation provided in Alg. 2, the risk evaluation function includes a parameter λ . This parameter determines how much the system is sensitive to disturbances and uncertainties. The higher value for this parameter means the graph nodes for the PRM should be selected more conservatively and the lower value means the closed-loop system is robust and the path planning can be less conservative. Therefore, it is necessary to do a statistical analysis of the closed-loop system behaviour to determine a proper value for this coefficient. In the simulation result section, the statistical analysis conducted for this reason is presented. The risk assessment functions obtained in Alg.2, are used to generate a search graph in the mission area. According to [41], different PRM algorithms apply different sampling techniques to increase the performance of the search graph in different situations. In this paper, the least amount of available data from the environment is used to determine weight functions according to the collision risk and the proximity to the obstacle's boundary to generate the PRM graph's nodes. In this paper the suggested robust planning algorithm has three main steps to determine an effective search graph for a coverage mission.

- Generating an initial node set using a weighted sampling method
- Connecting the nodes via PRM criteria and neighbourhood conditions
- Checking the graph connectivity and adding nodes and connections to obtain a fully-connected graph

Alg.3 illustrates the procedure to generate the initial set of nodes for the search graph. The parameter P_{adm} shows the maximum allowable value for the risk evaluation function to accept a node in the final search graph

The next step is to use the PRM policy to generate the link between the nodes. To apply the PRM approach, it is necessary to define a condition to determine the neighbour nodes inside

Algorithm 3 Graph Node Generation Subroutine

Data: $\mathcal{P}(\mathcal{O}^b)$, \mathcal{E} , \mathcal{O}^b - risk assessment function, workspace boundaries, and obstacles' boundaries prior information
Result: \mathcal{N}^{PRM} - graph node set

```

i  $\leftarrow 1$ 
while  $i \leq n_{PRM}^{nodes}$  do
     $p^{temp} \leftarrow$  random point  $\in \mathcal{E}$ 
    for all  $\mathcal{O}_k^b$  do
        if  $p^{temp} \notin \mathcal{E}_o$  then
            |  $P_k(p^{temp}) \leftarrow \mathcal{P}_k(\mathcal{O}_k^b)$ 
        else
            |  $P_k(p^{temp}) \leftarrow 1$ 
     $\mathcal{O}_{P_{max}}^b, P_{max} \leftarrow \max\{P(p^{temp})\}$ 
    if  $p_{prm}(z) \geq h_{max}(\mathcal{O}_{P_{max}}^b)$  then
        |  $\mathcal{J}(p^{temp}) \leftarrow 1$ 
    else
        if  $P_{max} \leq P_{adm}$  then
            |  $\mathcal{J}(p^{temp}) \leftarrow 1$ 
        else
            |  $\mathcal{J}(p^{temp}) \leftarrow 0$ 
    if  $\mathcal{J}(p^{temp})$  then
        |  $\mathcal{N}^{PRM}\{i\} \leftarrow p^{temp}$ 
        |  $i \leftarrow i + 1$ 

```

where n_{PRM}^{nodes} is the number of nodes that should be generated for the graph.

the \mathcal{N}^{PRM} . In this paper, the Euclidian norm between two points is used as a criterion to check the neighbourhood condition between every two points in the graph. The following equation presents the neighbourhood condition for each node in the search graph:

$$\mathbb{N}_i = \{\forall p_j \in \mathcal{N}^{PRM} | j \neq i, \|p_i - p_j\| \leq D_{nei}^{PRM}\}. \quad (10)$$

Moreover, (11) defines the collision function between the line connecting a point in the search graph to each of its neighbours and the peripheral ellipses enclosing the obstacles' boundaries.

$$Cols_{i,j}^k = \begin{cases} 1 & \text{if } \mathcal{L}(p_i, p_j) - f_k^{el} = 0 \text{ has a solution} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where, $\mathcal{L}(p_i, p_j)$ represents the line passes from p_i and p_j . Also, it is essential to make a limited number of links for each node. Alg. 4 presents the procedure utilized to generate the search graph links according to PRM criteria [25].

In this stage, there is a search graph that includes random points covering the mission area. Although increasing the number of points in the graph might improve the coverage, the search speed decreases significantly due to computational complexity and moreover, there would be no guarantee for the search graph to be fully connected. Therefore, it is necessary to use a policy to establish the connectivity criteria and simultaneously keep the graph complexity as low as possible. This paper suggests a recursive method to use a low number of initial nodes and links and create a fully-connected graph

Algorithm 4 Graph Link Generation Subroutine

Data: $\mathcal{P}(\mathcal{O}^b)$, \mathcal{O}^b , \mathcal{N}^{PRM} - risk assessment function, obstacles' boundaries prior information, graph node set
Result: $\mathcal{A}_{G_0}, \mathcal{W}_{G_0}$ - graph adjacency matrix, node connectivity condition number

```

 $\mathcal{A}_{G_0} \leftarrow [\mathbf{0}]_{n_{PRM}^{nodes} \times n_{PRM}^{nodes}}$ 
for  $i = 1 : n_{PRM}^{nodes}$  do
     $l_i \leftarrow \text{randperm}(\min(l(\mathbb{N}_i), L^{PRM}))$ 
     $\mathbb{N}_i^L \leftarrow \mathbb{N}_i\{l_i\}$ 
    for for all  $p_j \in \mathbb{N}_i^L$  do
        if  $\exists Cols_{i,j}$  or  $\mathcal{A}_{G_0}[i, j] \neq 0$  creates a loop
            then
                |  $\mathcal{A}_{G_0}[i, j], \mathcal{A}_{G_0}[i, j] = 0$ 
            else
                |  $\mathcal{A}_{G_0}[i, j], \mathcal{A}_{G_0}[i, j] = \|p_i - p_j\|_2$ 
            end
    end
     $\mathcal{W}_{G_0}\{p_i\} \leftarrow \frac{\sum(\mathcal{A}_{G_0}[i, :] \geq 0)}{\max(l_i, L^{PRM})}$ 
end

```

where l_i is the number of the eligible neighbourhoods for the i th node and L^{PRM} is the maximum admissible links that each node can make with its neighbour

by adding points and links in the area that causes the graph to be disconnected. This algorithm utilizes lemma 1 [42] to check the connectivity condition for the graph and \mathcal{W}_{G_0} to find the points in the graph with the least number of connections.

Lemma 1: If $\{I\}$ is a column vector with all elements equal to 1 then $\mathbf{0}$ is an eigenvalue of the Laplacian matrix \mathcal{L}_G with $\{I\}$ as a corresponding right eigenvector and all nonzero eigenvalues have positive real parts. Furthermore, zero is a simple eigenvalue of \mathcal{L}_G if and only if there exists a directed spanning tree in G . For an undirected graph G , the smallest nonzero eigenvalue λ_2 of \mathcal{L}_G , satisfies the condition in (12) and we have [42]

$$\lambda_2 = \min_{x \neq 0, \{I\}^T x = 0} \frac{x^T \mathcal{L}_G x}{x^T x}. \quad (12)$$

According to Alg. 5, the recursive policy adds extra nodes and links to the search graph until it becomes connected and satisfies the PRM conditions for finding at least one path between every two points in the graph [25]. The search graph produced with Alg. 2 to Alg. 5, creates a set of safe routes for the quadrotor to pass the environment without hitting the obstacle. However, accomplishing a coverage mission requires a set of target points in the mission area, which guarantees complete coverage. In this paper, it is assumed that according to the sensors used for data collection, the quadrotor is able to collect data from its neighbourhood that includes a spherical domain with R_s radius. Therefore, applying a proper classification algorithm that considers the sensor limits results in dividing the graph nodes into a few distinct sets. Each set has a unique central point that could be considered as a target

Algorithm 5 Recursive Algorithm Generating a Fully Connected Search Graph

Data: $\mathcal{N}^{PRM}, \mathcal{A}_{\mathcal{G}_0}, \mathcal{W}_{\mathcal{G}_0}$ - Initial graph node set, graph adjacency matrix, node connectivity condition number
Result: \mathcal{G}^{C-PRM} - Connected PRM graph

```

 $\mathcal{A}_{\mathcal{G}}, \mathcal{W}_{\mathcal{G}} \leftarrow \mathcal{A}_{\mathcal{G}_0}, \mathcal{W}_{\mathcal{G}_0}$ 
while  $\lambda_2\{\mathcal{L}_{\mathcal{G}}\} < 1$  do
   $k_{weak} \leftarrow \text{argmin}\{\mathcal{W}_{\mathcal{G}}\}$ 
   $D_{min} \leftarrow \min\{\|p_{k_{weak}} - p_i\|, i \in \mathcal{N}^{PRM} \& i \neq k_{weak}\}$ 
   $\mathcal{E}_{k_{weak}} \leftarrow \{\forall p \in \mathcal{E} \mid \|p - p_{k_{weak}}\| \leq \frac{1}{2}D_{min}\}$ 
  Update  $\mathcal{N}^{PRM} \leftarrow \text{Alg.3}$  with  $\mathcal{E}_{k_{weak}}$  and  $n_{nig}^{aug}$ 
  Update  $\mathcal{A}_{\mathcal{G}}, \mathcal{W}_{\mathcal{G}} \leftarrow \text{Alg.4}$ 
  Update  $\mathcal{L}_{\mathcal{G}} \leftarrow \mathcal{A}_{\mathcal{G}}$ 
end
 $\mathcal{G}^{C-PRM} \leftarrow \{\mathcal{N}^{PRM}, \mathcal{A}_{\mathcal{G}}\}$  where  $n_{nig}^{aug}$  is the number of augmented neighbours that should be added in the weak area to make connections between two possible separated subgraphs.
  
```

Algorithm 6 Sub-Domains and Target Points Determination

Data: \mathcal{G}^{C-PRM}, R_s - connected PRM graph, desired search radius
Result: $\mathcal{Q}^{tg}, \mathcal{G}^{C-PRM}$ - target point set, augmented PRM graph

```

 $\mathbb{A}^{tg} \leftarrow \text{UPGMA}(\mathcal{N}^{PRM}, R_s)$ 
for  $\forall \mathbb{A}^{tg} \in \mathbb{A}^{tg}$  do
   $| q_i^{tg} \leftarrow \text{mean}\{p \in \mathbb{A}^{tg}\}$ 
end
 $\mathcal{N}_{aug}^{PRM} \leftarrow \mathcal{N}^{PRM} \cup \mathcal{Q}^{tg}$ 
 $\mathcal{A}_{\mathcal{G}}^{tg} \leftarrow \text{Alg.4}$  with  $\mathcal{N}_{aug}^{PRM}$ 
 $\mathcal{G}^{R-PRM} \leftarrow \mathcal{A}_{\mathcal{G}}^{tg}, \mathcal{N}_{aug}^{PRM}$ 
  
```

point for the quadrotor to cover the whole sub-domain of the mission area. This paper applies the UPGMA algorithm to determine the sub-domains [43], [44]. Alg 6 describes the procedure to find the target points and the augmented graph $\mathcal{G}_{aug}^{C-PRM}$ which lets the quadrotor have access to the target points \mathcal{Q}^t .

B. Path Planning

This section presents the planning algorithm for the quadrotor. As stated in the second section, the planning algorithm should maximize the value function in (5). Therefore, the value function implies three necessary conditions to find a robust plan for a coverage mission due to the actuation system's limit for force and torque generation.

- The sequence of target points should result in a minimum travelling distance through the area
- The route utilized to arrive at each target point should satisfy the shortest path condition
- The trajectory that connects the waypoints in each route should satisfy the optimality condition

According to these three criteria, the first stage includes the evaluation of the best sequence for the target points obtained from the suggested robust PRM algorithm for area coverage. If the mission cost at the i_{th} step of the mission is stated as \mathbf{J}_i^C , the optimal sequence of the target points would be evaluated as a solution for the following problem:

$$\begin{aligned} \mathcal{Q}^{t^*} &= \underset{\mathcal{Q}^t}{\operatorname{argmin}} \left(\sum_{i=1}^{N_{tg}} \mathbf{J}_i^C \right), \\ \mathbf{J}_i^C &= \underset{\mathcal{Q}^t}{\min} \left(\sum_{j=1}^{N_{wp}^i-1} \|p_{j+1} - p_j\| \right), \end{aligned} \quad (13)$$

where, N_{wp}^i is the number of waypoints $\{p_k\} \subset \mathcal{N}_{aug}^{PRM}$ in the route between the $q_{l_i}^*$ and $q_{l_{i+1}}^* \in \mathcal{Q}^*$. Algorithm 7 presents the solving procedure of (13) in two step.

Algorithm 7 Shortest Path and Cost Between Target Points

Data: $\mathcal{G}_{aug}^{C-PRM}, Q^t, p_0$ - augmented PRM graph, target point set, initial position
Result: $\{\mathcal{P}ath_i^*\}, \mathcal{Q}^{t^*}$ - shortest path set, best ordered target point set

```

for  $i = 0 : N_{tg}$  do
  for  $j = 0 : N_{tg}, j \neq i$  do
     $| \mathcal{P}ath_{ij} \leftarrow \text{Dijkstra}(\mathcal{G}_{aug}^{C-PRM}, q_{l_i}, q_{l_j})$ 
     $| \mathbf{J}_{ij}^C \leftarrow \mathbf{D}(\mathcal{P}ath_{ij})$ 
  end
  end
   $\mathbf{TSM}(\{p_0, \mathcal{Q}^t\}, \{\mathbf{J}_{ij}^C\})$ 
   $\{\mathcal{P}ath_i^*\}, \mathcal{Q}^{t^*} \leftarrow \text{GA}(\mathbf{TSM})$ 
  
```

Dijkstra is a function that applies the Dijkstra method to find the shortest path between two nodes in a graph [45], and **D** represents a function that calculates the length of the path between two nodes in a graph. The combinatorial optimization results in \mathcal{Q}^{t^*} resembles a TSM¹ problem. The TSM problem associated with the proposed path planning in this paper is defined in three-dimensional space to consider the overall distance between the target points. There are many solvers to find a solution for a TSM problem, one of the fastest approaches is to use an evolutionary method like GA² [46]. Although Algorithm 7 determines the shortest path for the area coverage, it is necessary to minimize the search time to maximize the coverage score with respect to (5).

C. Trajectory Design

The set of waypoints determined from the motion planning algorithm cannot go through the control layer directly because the control law cannot handle discontinuities and it needs a continuous and smooth signal for the desired position and velocity vector at each time step. Therefore, another module

¹Traveling Salesman.

²Genetic Algorithm.

should make a bridge between the control and motion planning layers. In this section, an optimal trajectory design is proposed to convert the output of the path planning algorithm into a continuous set of signals for the desired trajectory and velocity considering the restrictions for acceleration. The following theorem demonstrates the resulting policy to determine the proper input signals for the control layer

Theorem 1: If the system described by (14) is considered as the kinematic model of the desired trajectory that passes the waypoints between each pair of target points in the mission area:

$$\begin{bmatrix} \dot{\xi}_d \\ \ddot{\xi}_d \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_d \\ \dot{\xi}_d \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u}_{tr}, \quad (14)$$

and (15) describes the objective function to minimize the time with respect to multiple terminal constraints and the actuator limits to generate desired acceleration for trajectory design:

$$\begin{aligned} \mathbf{J}^{Trj} &= \sum_{i=0}^{N_{wp}} (h_i(x_\xi(t_{f_i})) + \int_0^t \left(\frac{1}{2} \mathbf{u}_{tr}^T R_a \mathbf{u}_{tr} + 1 \right) d\tau), \\ x_\xi(t_f) &= \begin{bmatrix} \xi_d \\ \dot{\xi}_d \end{bmatrix} \in \mathbb{R}^6, R_a > 0 \in \mathbb{R}^{3 \times 3}, \\ h_i(x_\xi(t_{f_i})) &= \dot{\xi}_{d|p_i}, p_i \in Path_k. \end{aligned} \quad (15)$$

The Pontryagin's minimum principle [47] leads to the following solution for ξ_d^* , $\dot{\xi}_d^*$, and \mathbf{u}_{tr}^* for each segment of the

$$\begin{cases} \begin{bmatrix} \dot{x}_{\xi_1}^* \\ \dot{x}_{\xi_2}^* \\ \dot{\lambda}_{\xi_1}^* \\ \dot{\lambda}_{\xi_2}^* \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -R_a^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_{\xi_1} \\ x_{\xi_2} \\ \lambda_{\xi_1} \\ \lambda_{\xi_2} \end{bmatrix} \\ \mathbf{u}_{tr}^* = -R_a^{-1} \lambda_{\xi_2}^* \\ \lambda_{\xi_1}^* = 3R_a \left(\Delta \xi_{d|p_i, p_{i-1}} - \dot{\xi}_{d|p_{i-1}} \Delta t_{p_i, p_{i-1}} \right) \Delta t_{p_i, p_{i-1}}^{-3} \\ s_4 \Delta t_{p_i, p_{i-1}}^4 + s_3 \Delta t_{p_i, p_{i-1}}^3 + s_2 \Delta t_{p_i, p_{i-1}}^2 \\ + s_1 \Delta t_{p_i, p_{i-1}} + s_0 = 0 \end{cases} \quad (16)$$

where, $\lambda_{1\xi}$ and $\lambda_{2\xi}$ are Lagrangian multipliers, $\Delta t_{p_i, p_{i-1}}$ is the traveling time between p_i and $p_{i-1} \in Path_k$, $\Delta \xi_{d|p_i, p_{i-1}}$ is the relative vector between p_i and p_{i-1} , and s_0, \dots, s_4 are obtained from (17) as follows:

$$\begin{cases} s_0 = -9 \Delta \xi_{d|p_i, p_{i-1}}^T R_a \Delta \xi_{d|p_i, p_{i-1}} \\ s_1 = 6 \Delta \xi_{d|p_i, p_{i-1}}^T (-\mathbb{I}_3 + 3R_a) \dot{\xi}_{d|p_{i-1}} \\ s_2 = -\dot{\xi}_{d|p_{i-1}}^T (9R_a - 6\mathbb{I}_3) \dot{\xi}_{d|p_{i-1}} \\ s_3 = 0 \\ s_4 = 2 \end{cases}, \quad (17)$$

if the final velocity vector is free, and

$$\begin{aligned} \lambda_{\xi_1}^* &= 3R_a \left(\frac{9}{2} \Delta \xi_{d|p_i, p_{i-1}}^T R_a \Delta \xi_{d|p_i, p_{i-1}} \right)^{-\frac{3}{4}} \Delta \xi_{d|p_i, p_{i-1}} \\ \Delta t_{p_i, p_{i-1}} &= \left(\frac{9}{2} \Delta \xi_{d|p_i, p_{i-1}}^T R_a \Delta \xi_{d|p_i, p_{i-1}} \right)^{\frac{1}{4}} \end{aligned} \quad (18)$$

with the same state and control equation as (16) if the final velocity vector goes to zero.

Proof: Applying The principle of optimality, the main objective function can be written as a summation introduced in the following equation:

$$\mathbf{J}^{Trj} = \mathbf{J}_{p_{i-1}, p_i}^{Trj} + \mathbf{J}_{p_i, p_2}^{Trj} + \dots + \mathbf{J}_{p_{n_k-1}, p_{n_k}}^{Trj},$$

Accordingly each $\mathbf{J}_{p_{i-1}, p_i}^{Trj}$ should be optimised to comply with the main principle of optimality. The structure for each $\mathbf{J}_{p_{i-1}, p_i}^{Trj}$ is introduced in the following equation:

$$\mathbf{J}_{p_{i-1}, p_i}^{Trj} = h_i(x_\xi(t_{f_i})) + \int_{t_{i-1}}^{t_i} \left(\frac{1}{2} \mathbf{u}_{tr}^T R_a \mathbf{u}_{tr} + 1 \right) d\tau. \quad (19)$$

Considering the objective function in 19 and the state equation, the following equation describes the Hamiltonian function:

$$\mathcal{H}_\xi = 1 + \frac{1}{2} \mathbf{u}_{tr}^T R_a \mathbf{u}_{tr} + \lambda_{1\xi}^T x_{2\xi} + \lambda_{2\xi}^T \mathbf{u}_{tr}. \quad (20)$$

wher $\lambda_\xi = [\lambda_{1\xi}^T \lambda_{2\xi}^T]^T$ is a vector including the Lagrangian multipliers The Pontryagin minimum principle establishes that the optimal solution satisfies the following conditions:

$$\begin{cases} \dot{x}_\xi^* = \frac{\partial \mathcal{H}_\xi}{\partial \lambda_\xi} (x_\xi^*, \lambda_\xi^*, \mathbf{u}_{tr}^*, t) \\ \dot{\lambda}_\xi^* = \frac{\partial \mathcal{H}_\xi}{\partial x_\xi} (x_\xi^*, \lambda_\xi^*, \mathbf{u}_{tr}^*, t) \\ \frac{\partial \mathcal{H}_\xi}{\partial \mathbf{u}_{tr}} (x_\xi^*, \lambda_\xi^*, \mathbf{u}_{tr}^*, t) = 0 \end{cases} \quad (21)$$

Consequently, the Lagrangian multipliers and the control policy would be obtained from:

$$\begin{cases} \dot{\lambda}_{1\xi}^* = 0 \\ \dot{\lambda}_{2\xi}^* = \lambda_{1\xi}^* \\ \mathbf{u}_{tr}^* = -R_a^{-1} \lambda_{2\xi}^* \end{cases} \quad (22)$$

Furthermore, to achieve the final solution, the boundary conditions should be satisfied. the boundary condition equation is written as follows to comply with each waypoint p_i individually

$$\begin{aligned} &\left[\frac{\partial h_i}{\partial x_{1\xi}} (x_{\xi|p_i}^*, t_{p_i}) - \lambda_{1\xi|p_i}^* \right] \delta x_{1\xi|p_i} \\ &+ \left[\frac{\partial h_i}{\partial x_{2\xi}} (x_{\xi|p_i}^*, t_{p_i}) - \lambda_{2\xi|p_i}^* \right] \delta x_{2\xi|p_i} \\ &+ \left[\mathcal{H}_\xi (x_{\xi|p_i}^*, \mathbf{u}_{tr|p_i}^*, \lambda_{\xi|p_i}^*, t_{p_i}) \right. \\ &\left. + \frac{\partial h_i}{\partial t} (x_{\xi|p_i}^*, t_{p_i}) \right] \delta t_{p_i} = 0 \end{aligned} \quad (23)$$

where $\delta x_{1\xi|p_i}$, $\delta x_{2\xi|p_i}$, and δt_{p_i} are the various functions for the final states and time respectively. The final position vector at each stage is equivalent to one of the waypoints in the path. Therefore the variation function $\delta x_{1\xi|p_i}$ becomes zero automatically. Although there are no restrictions for the final velocity and reach time, it is necessary to vanish the velocity

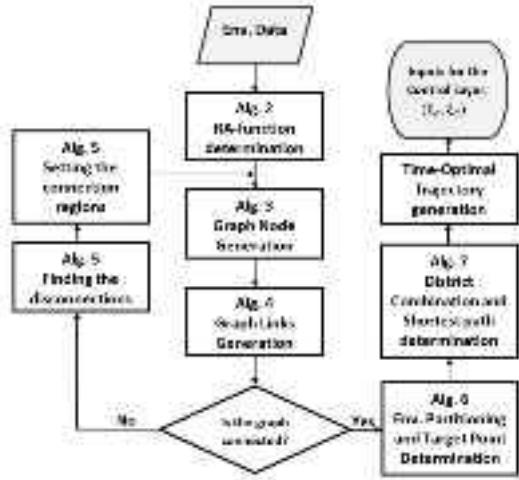


Fig. 2. Planning algorithm flowchart includes the loop to generate a connected graph, environment partitioning, and trajectory design.

vector and stop at the main target points. Accordingly, the boundary conditions can be summarized into three equations:

$$\begin{cases} x_{1\xi|p_i} = p_i \\ \lambda_{2\xi|p_i}^* = 0 & \text{if } x_{2\xi|p_i}^* \text{ is free} \\ \lambda_{1\xi|p_i}^{*T} x_{2\xi|p_i}^* = -1 \\ x_{1\xi|p_i} = p_i & \text{(if) } x_{2\xi|p_i}^* \text{ is zero} \\ x_{2\xi|p_i} = 0 \end{cases} \quad (24)$$

Using (22) and (23) together results in a solution for δt_{p_i} and Lagrangian multipliers. Therefore, for each pair of p_i and p_{i-1} , (14), (22), and (24) conclude to (16) which optimizes the $\mathbf{J}_{p_i, p_{i-1}}^{Trj}$. Consequently, The set of the solution can be considered as a solution for the optimization problem described by (15) and the Theorem1 will be proved.

Using the optimal policy obtained in Theorem1, and the waypoints from the planning algorithm, it is possible to search an uncertain environment with the least amount of prior information and make safe and smooth desired trajectories for the quadrotor to reach the target points and cover the mission area in minimum time with respect to actuator limits. Although the planning algorithm results in an optimal solution, it is necessary to use a control algorithm to guarantee robust performance in the presence of different uncertainties, exogenous disturbances, and measurement noise. The following section illustrates the NLH_∞ as a solution for this problem. The flowchart presented in Fig.2 demonstrates the relation between all the algorithms illustrated in this section.

IV. OUTPUT-FEEDBACK NLH_∞

According to [39], if the dynamic model of the quadrotor is stated as (3) in section II-B, and the effects of disturbances and modelling uncertainties are considered in d_ξ and d_η , a cascade NLH_∞ controller could stabilize the system and guarantee the boundedness of the outputs through time. As described in Fig. 1, The suggested control scheme includes a cascade architecture for position tracking and dynamic stabilization.

Accordingly, a desired control force vector will be obtained by (25) including parts to eliminate nonlinear elements, guarantee the tracking error, and make the system robust against uncertainties and disturbances.

$$\begin{aligned} f_d^c = & m\ddot{\xi} + K_\xi \dot{\xi} + mG \\ & - T_{1\xi}^{-1}(mT_\xi \dot{e}_\xi + K_\xi T_\xi e_\xi) \\ & + T_{1\xi}^{-1}u_\xi, \end{aligned} \quad (25)$$

where $T_\xi = [T_{1\xi} \ T_{2\xi} \ T_{3\xi}] \in \mathbb{R}^{3 \times 9}$ includes three gain matrices regarding the dynamic equation that describes the time-differential equation for the trajectory error $e_\xi = \begin{bmatrix} \dot{\xi} - \dot{\xi}_d \\ \ddot{\xi} - \ddot{\xi}_d \\ f(\xi - \xi_d) \end{bmatrix} \in \mathbb{R}^9$. Utilizing the suggested robust output-feedback control policy in [39], the following constitution would be satisfied automatically:

$$\int_0^t \|z_\xi\|_2^2 d\tau \leq \int_0^t \left\| \begin{bmatrix} w_\xi \\ v_\xi \end{bmatrix} \right\|_2^2 d\tau, \quad (26)$$

where w_ξ and v_ξ are determined as the modelling uncertainties and the measurement noise respectively, and z_ξ is defined as the objective function related to the following system:

$$\begin{aligned} \dot{e}_\xi &= f_\xi(e_\xi, t) + g_{1\xi}(e_\xi, t) \begin{bmatrix} w_\xi \\ v_\xi \end{bmatrix} + g_{2\xi}(e_\xi, t)u_\xi, \\ z_\xi &= \mathcal{W}_\xi \begin{bmatrix} e_\xi \\ u_\xi \end{bmatrix}, \text{ where } \mathcal{W}_\xi^T \mathcal{W}_\xi = \begin{bmatrix} Q_\xi & \mathbf{0} \\ \mathbf{0} & R_{u_\xi} \end{bmatrix}, \\ y_\xi &= h_\xi(e_\xi, t) + k_{y_\xi} \begin{bmatrix} w_\xi \\ v_\xi \end{bmatrix}, \end{aligned} \quad (27)$$

where $y_\xi = \begin{bmatrix} \mathbb{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_3 & \mathbf{0} \end{bmatrix} e_\xi \in \mathbb{R}^6$. The detailed explanation of (27) and the design procedure for both the outer-loop and inner-loop controller is stated in [39]. According to (27) and the statements in [39], if $\xi_d, \dot{\xi}_d, w_\xi, v_\xi$ belong to $L_2(0, \infty)$, the robust performance would be guaranteed by the suggested control policy u_ξ . Therefore, the resulting desired control force would be obtained by the following control law which consists of three proportional, differential, and integral gain matrices obtained regarding the robust criteria stated before.

$$\begin{aligned} f_d^c = & m\ddot{\xi}_d + K_\xi \dot{\xi}_d + mG \\ & - m \left(K_p \hat{e}_\xi + K_d \dot{\hat{e}}_\xi + K_I \int (\hat{e}_\xi) dt \right), \\ \dot{\hat{\xi}} &= \dot{\xi}_d + \hat{e}_\xi, \end{aligned} \quad (28)$$

where \hat{e}_ξ is the estimated vector of e_ξ obtained from the following equation:

$$\begin{aligned} \dot{\hat{e}}_\xi = & f_\xi(\hat{e}_\xi, t) + g_{1\xi}(\hat{e}_\xi, t) \begin{bmatrix} \hat{w}_\xi \\ \hat{v}_\xi \end{bmatrix} \\ & + g_{2\xi}(\hat{e}_\xi, t)u_\xi \\ & + G_\xi(y_\xi - h(\hat{e}_\xi, t)), \end{aligned} \quad (29)$$

and the gain matrices for both control and estimation parts are represented in detail in our previous work [39].

TABLE I
SIMULATION PARAMETER, UNCERTAINTY AND DISTURBANCES

Parameter	Definition	Value
m	robot's mass	0.75 kg
σ_m	mass uncertainty	30 %
I_{xx}	Moment of inertia about x axis	4e-3 kg.m ²
I_{yy}	Moment of inertia about y axis	4e-3 kg.m ²
I_{zz}	Moment of inertia about z axis	8e-3 kg.m ²
I_{xy}	cross moment of inertia	1e-3 kg.m ²
σ_I	Inertia uncertainty	30 %
K_ξ	Translational drag coefficient	0.01
K_η	Rotational drag coefficient	0.01
σ_K	Drag coefficient uncertainty	50 %
d_ξ	Disturbance force	$\mathcal{N}(0.5N, 0.3)$
d_η	Disturbance torque	$\mathcal{N}(0.05Nm, 0.05)$
σ_{obs}	Obstacles' boundary uncertainty	20 %

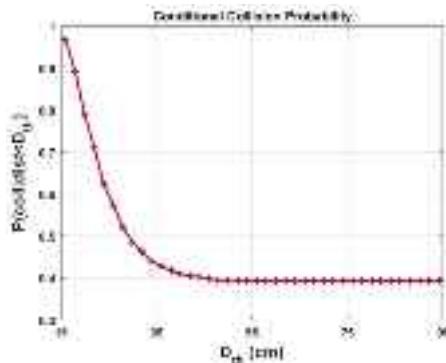


Fig. 3. Collision conditional probability in random paths obtained from the first-stage Monte Carlo simulation.

V. SIMULATION AND EXPERIMENTAL RESULTS

Two simulation techniques are used to evaluate the algorithm's performance. In the first step, a Monte Carlo simulation was conducted by applying 500 different sets of parameters and environmental conditions to establish robust performance via statistical analysis. Furthermore, the PX4 recommended software in the loop environment in conjunction with Gazebo physical simulation engine was utilized to evaluate the algorithm's capability for practical applications. The experiments were conducted based on the architecture presented in [39].

A. Monte-Carlo Simulation

In the Monte-Carlo simulation, 500 different samples were generated by applying different parameter sets to examine the robustness against parametric uncertainties and different random signals for measurement noise and disturbances to determine closed-loop dynamic behaviour. The uncertain parameters and the disturbances are introduced in Table I.

The Monte-Carlo simulations are conducted in two stages. In the first stage, a set of random paths through the environment were generated and the collision probability is obtained for all the samples. According to the statistical data, Fig. 3 shows the inverse relationship between the collision probability and the distance from the obstacles. The presented graph is used to determine the value for λ in the risk evaluation

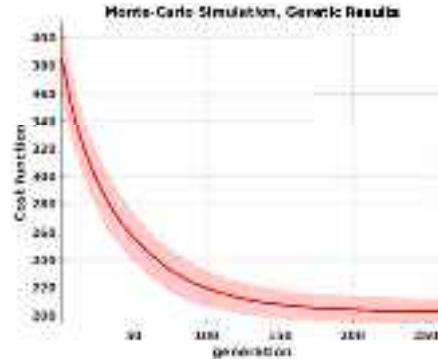


Fig. 4. Genetic Optimization results obtained from the 2nd stage Monte-Carlo simulation.

function stated in Alg. 2. Furthermore, the obtained result for the collision risk without the proposed motion planning establishes the necessity of using a proper motion algorithm in conjunction with a robust controller to maximise the mission success rate.

In each sample, a search graph is generated by the proposed algorithm. Then the path will be calculated based on Genetic optimization to obtain the best combination of target points and the time-optimal policy to determine the desired trajectory. The output-feedback NLH_∞ guarantees robust performance while the platform follows the desired trajectory afterwards. This section presents the statistical results obtained from the Monte-Carlo simulation. Figure 4 demonstrates the Genetic optimization results for the planning cost function. According to Fig. 4, it is clear that the suggested scheme for the optimization converged for all the samples in almost 250 generations, and there is a slight variation bound around the average line for the cost function. It is clear that despite the various search graphs and target point sets in each Monte-Carlo sample, the optimization remained in a narrow bound regarding the average line, which establishes the robustness of the proposed algorithm doing the optimization as the main part of the planning algorithm. Two other criteria are used to evaluate the variation in the planning results through the Monte-Carlo simulations. Figure 5 presents the overall travelling distance and time duration as a probability distribution graph. According to the statistical results depicted in Fig. 5, it is evident that applying the suggested algorithm to find an area coverage plan for an arbitrary environment obtains similar results for the total travelling distance and time despite all the uncertainties and random variations in the search graph. The ISE³ index was used to evaluate the closed-loop response and the trajectory tracking accuracy. Figure 6 depicts the ISE index for the tracking error in X and Y directions. According to Fig. 6, the tracking error in the XY-plane remains bounded with an almost normal distribution. It shows the controller can keep the quadrotor on the desired trajectory and make precise tracking. On the other hand, the IADU⁴ was utilized to evaluate the required control effort to remain robust against the environmental disturbances as modelled in (3) and reduce

³Integrated Squared error.

⁴Integral of the Absolute value of the control Derivative.

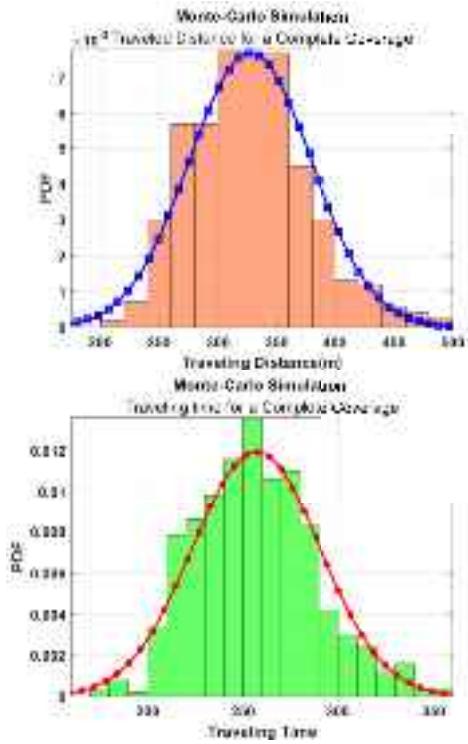


Fig. 5. Probability distribution for (top) the travelling distance and (bottom) time obtained from the 2nd stage Monte Carlo simulation.

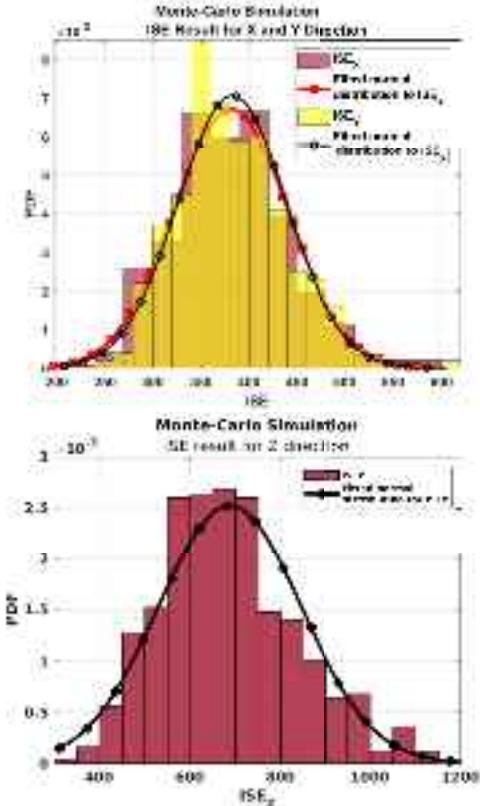


Fig. 6. Probability distribution for the trajectory ISE index in X, Y and Z directions.

the tracking error as much as possible. Figure 7, shows the probability distribution of the IADU index obtained by Monte-Carlo simulation: It is clear from Fig. 7 that the control

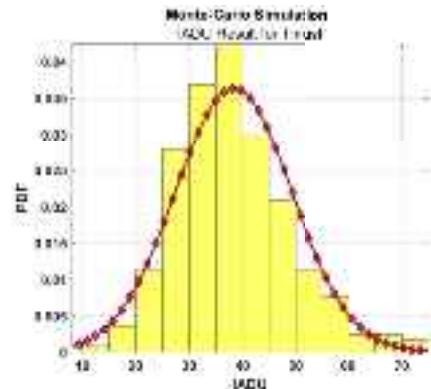


Fig. 7. Probability distribution for the IADU index.

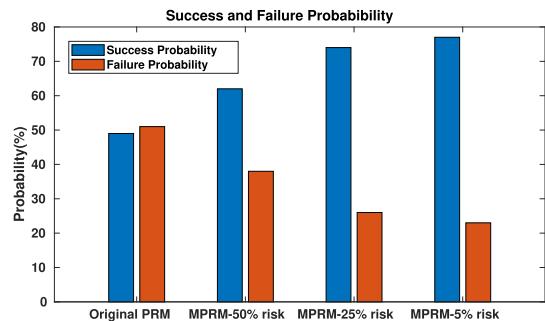


Fig. 8. Performance comparison between the integrated control-planning algorithm and the non-integrated planning with the original PRM.

effort remains bounded during the mission. Therefore, it can be concluded that the output-feedback NLH_{∞} is capable to guarantee the robustness of the suggested planning-control algorithm for an arbitrary area coverage mission.

Another Monte-Carlo analysis was conducted to compare the coverage mission's success rate in an environment including obstacles, disturbance forces and torques, and measurement noise considering the original PRM algorithm introduced in [41] for planning in a non-integrated coverage algorithm, and the proposed integrated robust planning algorithm with NLH_{∞} and genetic optimization to show the effectiveness of the suggested method. In these tests, each set includes 100 independent coverage simulations with the mentioned assumptions. Figure 8, presents the probability of success and failure as a criterion for comparing the performance of the proposed algorithm with different allowable risks for the mission and original form of the PRM algorithm. It is clear from Fig. 8 that the suggested integrated algorithm improved the success rate in an arbitrary coverage mission including obstacle avoidance and confronting disturbances and uncertainties. Moreover, that analysis demonstrates how the cascade architecture improves the performance of the planning algorithm compared to the case that the planning algorithm is not influenced by the control algorithm. Furthermore, it is evident that the algorithm can not eliminate the collision risk and make a 100% success rate even by reducing the allowable risk threshold in the algorithm. There are two main reasons for such results. First, the suggested algorithm applied the risk

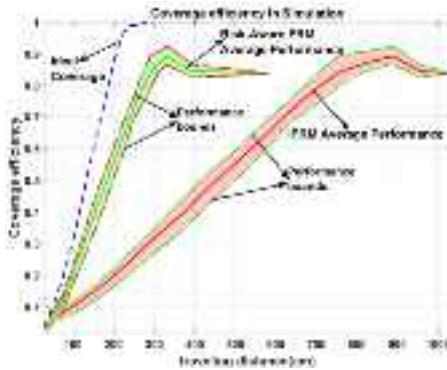


Fig. 9. Coverage performance comparison for proposed algorithm, Ideal coverage, and IPRM algorithm [48].

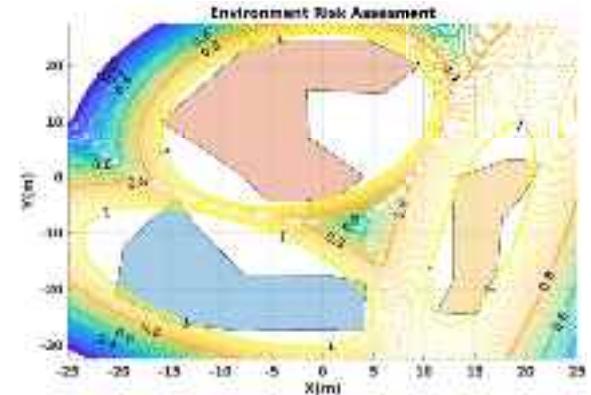


Fig. 10. Results for the risk assessment in XY-plane.

assessment procedure only in the graph node generation stage and accordingly, it is possible for the desired trajectory to pass through high-risk areas between two safe nodes. Also, the disturbance strength might exceed the actuation system limits and eventually, it causes failure and collision even if the flight path doesn't pass the high-risk districts. In future works, it is intended to add the risk assessment attribute to the graph links generation stage for further failure risk reduction.

Although Fig. 8 established the improvement of the proposed algorithm in comparison with the original PRM method, it is necessary to demonstrate the performance of the algorithm relative to other methods regarding general criteria to show the coverage performance in addition to the success rate. Therefore, two other coverage scenarios were considered for that reason. In the first scenario, the environment is completely known and there is no uncertainty or disturbances to make the failure risk. Furthermore, the exploration path is calculated by GA optimisation to make sure about achieving the minimum travelling distance for the coverage. In the second scenario, the IPRM algorithm suggested by [48] is applied for the exploration and the same conditions as considered for the proposed algorithm are applied for the statistical analysis. The performance criteria for the comparison is the coverage efficiency described by the ratio between the effective covered and the total workspace area. The effective covered area is obtained by subtracting the repeated covered area from the total covered area. The comparison result is demonstrated in Fig. 9.

According to the graphs presented in Fig. 9, The average performance obtained for the proposed algorithm resembles the ideal coverage performance while the implementation includes disturbances and uncertainties for the proposed algorithm. Furthermore, the variation bound is narrow relative to the average performance which establishes the algorithm's robustness against uncertainties and disturbances. It is evident that the proposed algorithm outperforms the results obtained from the IPRM algorithm for the exploration in an uncertain environment.

B. Software In the Loop

It is necessary to evaluate the algorithm's capability for hardware implementation and use in practical cases. In this

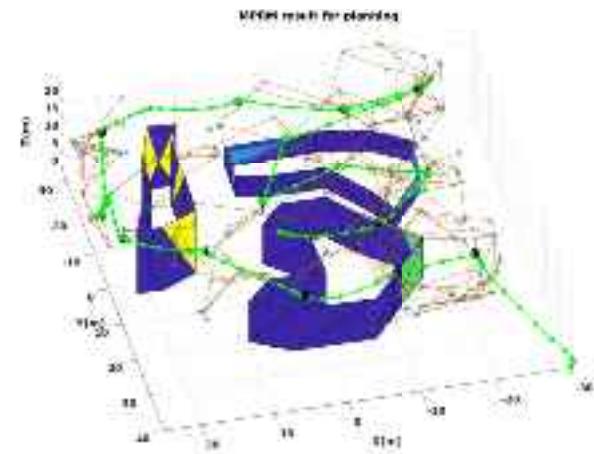


Fig. 11. MPRM planning results: blue stars: Search graph nodes, red lines: search graph links, black stars: target points for coverage, green dotted line: quadrotor's trajectory.

paper, the suggested PX4 SIL test environment including the Gazebo physical engine, MAVROS communication protocol, and PX4 flight management core for driving the sensors and actuators is utilized in order to establish the performance of the algorithm in a high-fidelity simulation environment. The main architecture for the experiment was demonstrated and verified in [39]. Figure 10 presents a sample result for the risk assessment of an arbitrary environment including three obstacles. As stated before, the information about the obstacles' boundaries is uncertain. According to Fig. 10, it is possible to limit the allowable risk threshold to generate the random points for the search graph and prevent the quadrotor to generate a path too close to the obstacles' boundaries and control the collision risk. Figure 11, depicts the planning results for an arbitrary environment. The red lines and the blue stars present the search graph obtained from Algorithm 1, the black stars represent the target points for the coverage obtained from Algorithm 6, and the green dotted line depicts the trajectory followed by the quadrotor during the mission. Figure 12 shows the system's closed-loop performance as the quadrotor's position vector components through time.

According to Fig. 12, it is evident that the quadrotors followed the desired trajectory precisely during the mission and the estimation part of the output-feedback NLH_∞ estimated

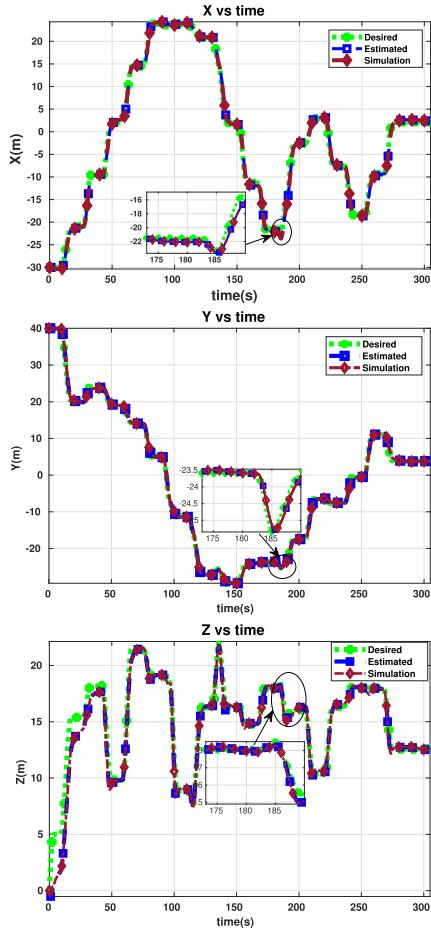


Fig. 12. Closed-loop result: quadrotor's position vector, Green line: Desired trajectory from optimal trajectory generation, Blue line: Estimated state from the control policy, Red line: the simulation result from Gazebo.

the position vector components from the measurement data accurately. Furthermore, that figure demonstrates the analysis results for the second layer's performance in the planning-control hierarchy. According to the presented result, the suggested algorithm is capable of making a desired trajectory smooth and continues according to the control policy requirements and there is no sharp edge indicating high acceleration even at waypoints' switching moments which are magnified in Fig. 12. Therefore, the results obtained from the simulation establish that the planning algorithm which depends on the closed-loop performance according to the risk-assessment function obtained from the Monte Carlo analysis can generate a continuous and smooth input for the controller. Although the tracking error for the z-component is more than the expected settling error in the first 35 s because of the limits considered for the vertical acceleration and velocity, the controller reduces the error to the acceptable bound and keeps it there till the end of the mission. However, it is necessary to evaluate the controller performance in terms of the desired attitude and the thrust level to see the required control effort for such a mission and if it can be considered a practical mission for a real quadrotor or not. Figure 13, presents the results for the desired attitude (roll and pitch angles) calculated by the control algorithm and the values obtained

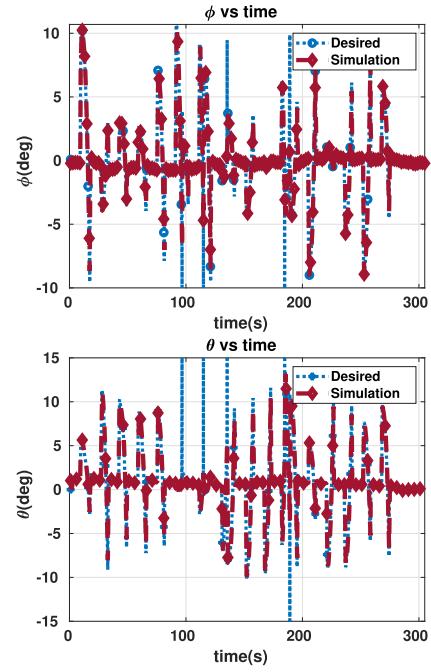


Fig. 13. Roll and Pitch angles, desired values obtained from the controller and the Gazebo simulation results.

from the Gazebo simulation. As depicted in Fig. 13, it is clear that the mid-level control commands in terms of desired roll and pitch angles were obtained in a reasonable range between -15° and 15° which is acceptable for a real drone to implement in the real flight test. Furthermore, a video was captured from this experiment and was placed in the following link: https://www.youtube.com/watch?v=9p0o8px7L_8

The results presented in this section establish the capability of the planning algorithm to determine a robust plan and optimal trajectory in order to accomplish a coverage mission in an arbitrary area. Furthermore, the closed-loop response of the system utilizing the output-feedback NLH_∞ establishes the robust performance of the system in the presence of uncertainties and disturbances.

C. Real-Robot Experiment

Although the SIL tests are conducted to establish the practical capability of the algorithm, real-robot experiments are also implemented to evaluate the algorithm's performance in the real world. For the real-robot experiment, the Crazyflie drone is used. That micro-drone is developed as an experimental setup to facilitate the flight test for newly developed algorithms [49]. To accomplish the flight experiments, a ROS bridge is made to make a reliable connection between the node which implements the algorithm and the flight management computer. The crazyflie-lib-python is used to make the bridge to control the platform. The workspace is a $2 \times 2 \times 2 \text{ m}^3$ space including two arbitrary obstacles placed in the workspace. Figure 14 demonstrates four different moments of the flight experiment conducted in the workspace. The video for the experiment is available on: <https://youtu.be/emCnWHNBfR8>.

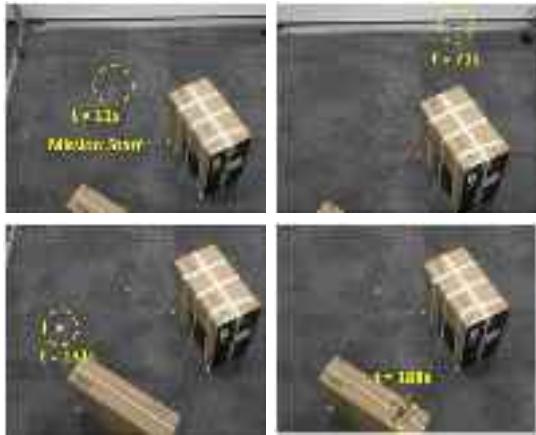


Fig. 14. Four snapshots at different times from the flight test implemented using a Crazzyflie micro-drone.

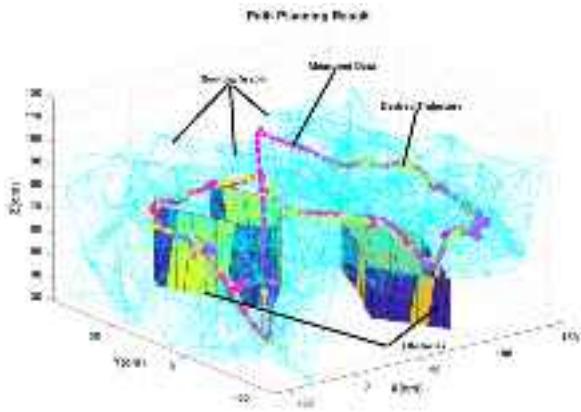


Fig. 15. Flight test result: Planning result and the desired trajectory through the environment.

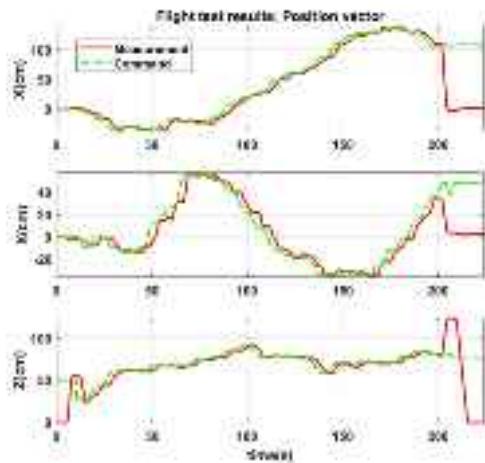


Fig. 16. Flight test data: position vector's components. The measurements obtained from the Crazzyflie lighthouse localisation system.

The conducted experiment resulted in six different districts for coverage and 27 waypoints to reach those districts. The planning result for that experiment is presented in Fig. 15.

Furthermore, the flight data acquired from the experiment is provided in Fig. 16. As demonstrated in that figure, the robot followed the desired trajectory closely despite all the communication delays.

VI. CONCLUSION

This paper suggested a novel unified planning-control algorithm based on the proposed robust PRM method and the output-feedback NLH_{∞} theory. The proposed algorithm applies a joint architecture to exploit the robust feature of the NLH_{∞} in conjunction with optimal trajectory design and probabilistic planning to generate a uniform, comprehensive policy to accomplish an area coverage mission. Although the theoretical approach demonstrates the robustness and efficiency characteristic of the algorithm, a Monte-Carlo simulation has been conducted to evaluate the system's performance using a statistical approach. According to the 500 samples with different parameter sets, measurement noise, and disturbance signals, the optimization would converge during almost 250 generations and the reducing cost function shows a slight change around the average line which means that the suggested algorithm guarantees finding an optimal solution and this solution would be robust against the uncertainties. The sharp probability distribution functions obtained for the total travelling distance and time establish that the system's performance is almost the same in the majority of samples. Moreover, the statistical analysis comparing the proposed cascade framework and the PRM algorithm working independently from the control system demonstrates how the cascade architecture improves performance and decreases the failure risk. On the other hand, the SIL experiments establish the capability of the planning and control algorithm to accomplish a complete area coverage mission in a realistic environment by applying reasonable control efforts in terms of the desired attitude and the throttle level. It can be concluded that the proposed algorithm is a compatible combination of planning and control algorithms that are adjusted to make a comprehensive area coverage management policy.

REFERENCES

- [1] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1258–1276, Dec. 2013.
- [2] L. Paull, M. Seto, J. J. Leonard, and H. Li, "Probabilistic cooperative mobile robot area coverage and its application to autonomous seabed mapping," *Int. J. Robot. Res.*, vol. 37, no. 1, pp. 21–45, Jan. 2018.
- [3] K. Wu, J. Hu, Z. Ding, and F. Arvin, "Finite-time fault-tolerant formation control for distributed multi-vehicle networks with bearing measurements," *IEEE Trans. Autom. Sci. Eng.*, early access, Jan. 30, 2023, doi: 10.1109/TASE.2023.3239748.
- [4] B. K. Patle, G. Babu L. A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technol.*, vol. 15, no. 4, pp. 582–606, Aug. 2019.
- [5] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, vol. 9, pp. 119310–119342, 2021.
- [6] C. Dimidov, G. Oriolo, and V. Trianni, "Random walks in swarm robotics: An experiment with kilobots," in *Proc. Int. Conf. Swarm Intell.*, 2016, pp. 185–196.
- [7] J. Xie, L. R. Garcia Carrillo, and L. Jin, "Path planning for UAV to cover multiple separated convex polygonal regions," *IEEE Access*, vol. 8, pp. 51770–51785, 2020.
- [8] B. J. Englot and F. S. Hover, "Sampling-based coverage path planning for inspection of complex structures," in *Proc. 22nd Int. Conf. Automated Planning Scheduling*, 2012, pp. 29–37.
- [9] C.-H. Li, C. Fang, F.-Y. Wang, B. Xia, and Y. Song, "Complete coverage path planning for an Arnold system based mobile robot to perform specific types of missions," *Frontiers Inf. Technol. Electron. Eng.*, vol. 20, no. 11, pp. 1530–1542, Nov. 2019.

- [10] S. Xie, J. Hu, P. Bhowmick, Z. Ding, and F. Arvin, "Distributed motion planning for safe autonomous vehicle overtaking via artificial potential field," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21531–21547, Nov. 2022.
- [11] W. Dong, S. Liu, Y. Ding, X. Sheng, and X. Zhu, "An artificially weighted spanning tree coverage algorithm for decentralized flying robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1689–1698, Oct. 2020.
- [12] K. R. Guruprasad and T. D. Ranjitha, "CPC algorithm: Exact area coverage by a mobile robot using approximate cellular decomposition," *Robotica*, vol. 39, no. 7, pp. 1141–1162, Jul. 2021.
- [13] Á. Madridano, A. Al-Kaff, D. Martín, and A. de la Escalera, "3D trajectory planning method for UAVs swarm in building emergencies," *Sensors*, vol. 20, no. 3, p. 642, 2020.
- [14] C. Wei et al., "Research on UAV intelligent obstacle avoidance technology during inspection of transmission line," in *Proc. Appl. Mech., Mechatronics Intell. Syst.*, Feb. 2016, pp. 319–327.
- [15] S. Godio, S. Primatesta, G. Guglieri, and F. Dovis, "A bioinspired neural network-based approach for cooperative coverage planning of UAVs," *Information*, vol. 12, no. 2, p. 51, Jan. 2021.
- [16] S. G. Potapova, A. V. Artemov, S. V. Sviridov, D. A. Musatkina, D. N. Zorin, and E. V. Burnaev, "Next best view planning via reinforcement learning for scanning of arbitrary 3D shapes," *J. Commun. Technol. Electron.*, vol. 65, no. 12, pp. 1484–1490, Dec. 2020.
- [17] Y. Tang, R. Zhou, G. Sun, B. Di, and R. Xiong, "A novel cooperative path planning for multirobot persistent coverage in complex environments," *IEEE Sensors J.*, vol. 20, no. 8, pp. 4485–4495, Apr. 2020.
- [18] A. Bircher et al., "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Auto. Robots*, vol. 40, no. 6, pp. 1059–1078, Aug. 2016.
- [19] S. Xie, J. Hu, Z. Ding, and F. Arvin, "Cooperative adaptive cruise control for connected autonomous vehicles using spring damping energy model," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 2974–2987, Mar. 2023.
- [20] J. Rao, B. Li, Z. Zhang, D. Chen, and W. Giernacki, "Position control of quadrotor UAV based on cascade fuzzy neural network," *Energies*, vol. 15, no. 5, p. 1763, Feb. 2022.
- [21] V. A. Laurence and J. C. Gerdes, "Long-horizon vehicle motion planning and control through serially cascaded model complexity," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 1, pp. 166–179, Jan. 2022.
- [22] J. Z. Lu, "Closing the gap between planning and control: A multi-scale MPC cascade approach," *Annu. Rev. Control*, vol. 40, pp. 3–13, Jan. 2015.
- [23] A. C. Manav and I. Lazoglu, "A novel cascade path planning algorithm for autonomous truck-trailer parking," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6821–6835, Jul. 2022.
- [24] R. Sepulchre, M. Jankovic, and P. V. Kokotovic, *Constructive Nonlinear Control*. Cham, Switzerland: Springer, 2012.
- [25] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
- [26] K. Cao, Q. Cheng, S. Gao, Y. Chen, and C. Chen, "Improved PRM for path planning in narrow passages," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2019, pp. 45–50.
- [27] M. I. Chowdhury and D. G. Schwartz, "The PRM-A* path planning algorithm for UUVs: An application to navy mission planning," in *Proc. Global Oceans 2020: Singapore U.S. Gulf Coast*, Oct. 2020, pp. 1–9.
- [28] A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "HPPRM: Hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots," *IEEE Access*, vol. 8, pp. 221743–221766, 2020.
- [29] M. Hoerger, "Tractable POMDP-planning for robots with complex non-linear dynamics," Ph.D. dissertation, School Inf. Technol. Elect. Eng., Univ. Queensland, Brisbane, QLD, Australia, 2020.
- [30] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 268–304, Feb. 2014.
- [31] J. P. van den Berg, D. Nieuwenhuisen, L. Jaillet, and M. H. Overmars, "Creating robust roadmaps for motion planning in changing environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 1053–1059.
- [32] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.
- [33] F. S. Barbosa, B. Lacerda, P. Duckworth, J. Tumova, and N. Hawes, "Risk-aware motion planning in partially known environments," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 5220–5226.
- [34] P. De Petris, M. Dharmadhikari, H. Nguyen, and K. Alexis, "Risk-aware motion planning for collision-tolerant aerial robots subject to localization uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 4561–4568.
- [35] K. Cai, C. Wang, S. Song, H. Chen, and M. Q.-H. Meng, "Risk-aware path planning under uncertainty in dynamic environments," *J. Intell. Robotic Syst.*, vol. 101, no. 3, pp. 1–15, Mar. 2021.
- [36] S. Feyzabadi and S. Carpin, "Risk-aware path planning using hierarchical constrained Markov decision processes," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2014, pp. 297–303.
- [37] A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme, "Risk-aware path planning for autonomous underwater vehicles using predictive ocean models," *J. Field Robot.*, vol. 30, no. 5, pp. 741–762, Sep. 2013.
- [38] T. Nyberg, C. Pek, L. Dal Col, C. Norén, and J. Tumova, "Risk-aware motion planning for autonomous vehicles with safety specifications," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 1016–1023.
- [39] F. Rekabi, F. A. Shirazi, M. J. Sadigh, and M. Saadat, "Nonlinear H_∞ measurement feedback control algorithm for quadrotor position tracking," *J. Franklin Inst.*, vol. 357, no. 11, pp. 6777–6804, Jul. 2020.
- [40] F. Kendoul, Z. Yu, and K. Nonami, "Guidance and nonlinear control system for autonomous flight of minirotocraft unmanned aerial vehicles," *J. Field Robot.*, vol. 27, no. 3, pp. 311–334, May 2010.
- [41] R. Geraerts and M. Overmars, "Sampling techniques for probabilistic roadmap planners," *Intell. Auton. Syst.*, pp. 600–609, 2004.
- [42] Z. Li, W. Ren, X. Liu, and L. Xie, "Distributed consensus of linear multi-agent systems with adaptive dynamic protocols," *Automatica*, vol. 49, no. 7, pp. 1986–1995, Jul. 2013.
- [43] C.-H. Lung and C. Zhou, "Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach," *Ad Hoc Netw.*, vol. 8, no. 3, pp. 328–344, May 2010.
- [44] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *Comput. J.*, vol. 26, no. 4, pp. 354–359, Nov. 1983.
- [45] B. V. Cherkassky, A. V. Goldberg, and T. Radzik, "Shortest paths algorithms: Theory and experimental evaluation," *Math. Program.*, vol. 73, no. 2, pp. 129–174, May 1996.
- [46] A. Oliynyk, I. Fedorchenko, A. Stepanenko, M. Rud, and D. Goncharenko, "Evolutionary method for solving the traveling salesman problem," in *Proc. Int. Sci.-Practical Conf. Problems Infocommunications. Sci. Technol. (PIC S&T)*, Oct. 2018, pp. 331–338.
- [47] D. E. Kirk, *Optimal Control Theory: An Introduction*. Chelmsford, MA, USA: Courier Corporation, 2004.
- [48] Q. Jin, Q. Hu, P. Zhao, S. Wang, and M. Ai, "An improved probabilistic roadmap planning method for safe indoor flights of unmanned aerial vehicles," *Drones*, vol. 7, no. 2, p. 92, Jan. 2023.
- [49] W. Giernacki, M. Skwierczynski, W. Witwicki, P. Wronski, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *Proc. 22nd Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2017, pp. 37–42.



Fatemeh Rekabi-Bana received the B.Sc. and M.Sc. degrees in aerospace engineering from the Amir Kabir University of Technology (Tehran Polytechnic) in 2010 and 2012, respectively, and the Ph.D. degree in mechanical engineering dynamics, control, and vibration from the University of Tehran in 2020. She joined as a Post-Doctoral Research Associate of the EU-H2020 Project (RoboRoyale) with The University of Manchester. She is currently a Post-Doctoral Research Associate with the Department of Computer Science.



Junyan Hu received the B.Eng. degree in automation from the Hefei University of Technology in 2015 and the Ph.D. degree in electrical and electronic engineering from The University of Manchester in 2020.

He is currently an Assistant Professor of robotics with the Department of Computer Science, Durham University. Prior to that, he was a Lecturer with University College London and a Post-Doctoral Research Associate with The University of Manchester. His current research interests include swarm intelligence, multi-agent systems, cooperative path planning, and distributed control, with applications to autonomous driving and robotics. He is a member of the IEEE-CSS Technical Committee on Networks and Communication Systems and the IEEE-RAS Technical Committee on Multi-Robot Systems. He served as an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS and IEEE International Conference on Robotics and Automation.



Farshad Arvin received the B.Sc. degree in computer engineering, the M.Sc. degree in computer systems engineering, and the Ph.D. degree in computer science in 2004, 2010, and 2015, respectively. He is currently an Associate Professor of robotics with the Department of Computer Science, Durham University, U.K. Prior to that, he was a Senior Lecturer of robotics with The University of Manchester, U.K. He visited several leading institutes, including the Artificial Life Laboratory, University of Graz; the Institute of Microelectronics, Tsinghua University, Beijing; and the Italian Institute of Technology (IIT), Genoa, as a Senior Visiting Research Scholar. He was the Founding Director of the Swarm and Computation Intelligence Laboratory (www.SwaCIL.com) formed in 2018. His current research interests include swarm robotics and autonomous multi-agent systems.



Tomáš Krajník is currently an Associate Professor with Czech Technical University in Prague and the Founding Director of the Chronorobotics Laboratory, which aims at the problems related to the long-term autonomy of mobile robots in naturally changing environments. He has participated in several EU and international projects and before returning to Czechia, he was a part of the Lincoln Centre for Autonomous Systems Research, led by Prof. Tom Duckett. His current research interests include long-term autonomy, spatiotemporal modeling, and swarm robotics.