Nonlinear Model Predictive Control for Repetitive Area Reconnaissance with a Multirotor Drone

Salvatore Marcellini, Fabio Ruggiero, Vincenzo Lippiello

***Abstract*— This paper considers the problem of a reconnaissance mission in which a single multirotor drone must survey a given map by repetitively visiting different checkpoints. Several points of interest (POIs) are used to discretise the map, and each of them is associated with a time-varying heat value according to the specific application. In that way, each POI has a different visiting priority each time. The proposed solution considers a nonlinear model predictive control (NMPC) approach that minimises the map's overall heat and considers several constraints related to the system dynamics and the environment (e.g., the presence of unknown obstacles). Possible applications are related to the research of gas leaks, area surveillance, patrolling, etc. The methodology is tested in a realistic simulation environment and through experiments.**

Video available at:
`https://youtu.be/IM8WvibB6Zo`

I. Introduction

Multirotor unmanned aerial vehicles (UAVs) nowadays are widely used in different domains thanks to the advancement in hardware technology and the increasing autonomy given by a more mature software [1]. Some application fields are inspection and maintenance [2], search and rescue [3], aerial delivery [4], and many more. In all the applications requiring a high level of autonomy, the UAV has either to find its way or safely navigate following a pre-defined trajectory to get to the target. For example, in [5], the drone can safely navigate in a subterranean environment following the commands given by a human operator. In [6], the authors developed a control structure to avoid dynamic obstacles while following a path. In other applications, the UAV does not have a specific reference point or target, and it is instead demanded to perform a task autonomously. In [7], the task consists in covering every point of a specific area of interest, while in [8], a group of drones have to survey a specific area. Because of the possibility to survey small and relatively large areas, the agility in overcoming obstacles, and the possibility to have a high viewpoint, UAVs are often used daily in surveillance tasks.

The surveillance task is often carried out with closed-circuit television cameras that continuously look at some areas of interest; however, the main problem with this approach is their placement [9]. Moreover, if there is a need to cover an ample space, it is not easy to monitor the whole area only by security staff. Instead, a UAV can provide a mobile viewpoint from above with a broader area coverage, partly solving the problem of the visual range occlusion [10], and the possibility of redirecting the survey to the desired point of interest (POI) [11].

In recent literature, the UAV is utilised as a flying camera for different surveillance tasks such as *area coverage* and *reconnaissance*. The most common one is the area coverage [7], where the agent has to follow a path, usually predefined, that maximises the area covered by the overseer. These paths depend on the geometry of the area that must be supervised [12] [13]. Area coverage has also been used in gas leaking research [14], [15]. In reconnaissance problems, instead, one or more UAVs are required to visit several checkpoints along their routes to the target location [16]–[18]. However, this approach leads to multiple problems, for example, in the *pursuit-evasion* problem [19], where the malicious entity that has to be found can predict the actions of the overseer. In pursuit-evasion problems, one or more searchers move throughout a given target area to guarantee the detection of all the evaders, which can move arbitrarily fast [20], and the searchers do not necessarily have to cover the entire target area, like in the first task.

This paper considers a single multirotor UAV (i.e., the *overseer*) in a reconnaissance mission that has to fly on a limited map, avoid collisions with unknown obstacles, and repeatedly visit some POIs. Possible applications are, for instance, the search for a moving intruder inside a designed area or gas leaks inside an industrial environment. The POIs are characterised by their coordinates and a heat value related to the application (e.g., the probability of an intruder's presence or the gas leak). This paper sees at the center of each POI a 2D Gaussian distribution that grows and expands while the overseer is near it, thus, the entire distribution can be seen as a *heat map*. This formulation can shape the POIs' heat value based on the current UAV location, the last time the UAV visited that POI, and external sensors or events that may give additional information about the POI. Notice that this reconnaissance mission is not interested in the complete area coverage but in the repetitive visit of the POIs by the overseer that must change its plan online depending on the evolution of the environment state. In order to solve this problem, this paper devises a nonlinear model predictive control (NMPC) that minimises the sum of the heat values taking into account (i) the UAV dynamic model, (ii) the distance from the various obstacles inside the map, and (iii) the evolution of the heat in each POI.

The research leading to these results has been supported by the AERIAL-CORE project (Horizon 2020 Grant Agreement No. 871479). The authors are solely responsible for its content.

The authors are with the PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, Naples, 80125, Italy. Corresponding author's e-mail: `salvatore.marcellini@unina.it`.

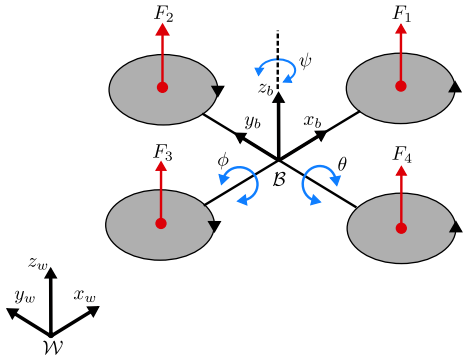


Fig. 1: UAV reference frames

The NMPC approach is well suited to solve the described problem because of its capability to adapt the solution to the changes in the problem state while satisfying some dynamic constraints. The contribution of this work lies in using the MPC technique to plan a route for the reconnaissance task. Thanks to the constraints, the path toward the different POIs does not consider a constant visiting order during the execution; instead, it changes with respect to the environment and a desired given POIs' behavior that can be time and state-dependent.

II. UAV DYNAMIC MODEL

The multirotor UAV coordinate systems are shown in Fig. 1, where $\mathcal{B} : (x^b, y^b, z^b)$ denotes the body fixed frame, attached to the UAV's center of mass, and $\mathcal{W} : (x^W, y^W, z^W)$ denotes the global world coordinate system. Let $\mathbf{p}_b = [x, y, z]^T \in \mathbb{R}^3$ be the position of the body frame $\mathcal{B}$ in the world frame $\mathcal{W}$, $\mathbf{R}_b \in SO(3)$ be the rotation matrix expressing the orientation of $\mathcal{B}$ with respect to $\mathcal{W}$, $\boldsymbol{\eta}_b = [\phi, \theta, \psi]^T \in \mathbb{R}^3$ be the roll-pitch-yaw Euler angles extracted from $\mathbf{R}_b$, $\dot{\mathbf{p}}_b = [v_x, v_y, v_z]^T \in \mathbb{R}^3$ be the UAV linear velocity, $\dot{\boldsymbol{\eta}}_b = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T \in \mathbb{R}^3$ be the time derivative of the Euler angles, and $\boldsymbol{\omega}_b^b \in \mathbb{R}^3$ be the angular velocity of the body frame with respect to the world frame expressed in the body frame. The relation between the time derivative of the Euler angles and the body angular velocity is dictated by the matrix $\mathbf{T}(\boldsymbol{\eta}_b) \in \mathbb{R}^{3\times3}$, such that $\boldsymbol{\omega}_b^b = \mathbf{T}(\boldsymbol{\eta}_b)\dot{\boldsymbol{\eta}}_b$.

The dynamic model of a UAV with Euler angles is [21]

$$\begin{cases} \ddot{\mathbf{p}}_b = -g\,\mathbf{e}_3 + \dfrac{1}{m} f_z\,\mathbf{R}_b\,\mathbf{e}_3, \\ \ddot{\boldsymbol{\eta}}_b = \mathbf{M}(\boldsymbol{\eta}_b)^{-1}(-\mathbf{C}(\boldsymbol{\eta}_b, \dot{\boldsymbol{\eta}}_b)\dot{\boldsymbol{\eta}}_b + \mathbf{T}(\boldsymbol{\eta}_b)\,\boldsymbol{\tau}_b), \end{cases} \tag{1}$$

where $\mathbf{M}(\boldsymbol{\eta}_b) = \mathbf{T}(\boldsymbol{\eta}_b)^T\,\mathbf{I}_b\,\mathbf{T}(\boldsymbol{\eta}_b) \in \mathbb{R}^{3\times3}$ is the symmetric and positive definite mass matrix, provided that $\theta \neq \pm\frac{\pi}{2}$ rad, $\mathbf{I}_b \in \mathbb{R}^{3\times3}$ is the diagonal and positive definite UAV inertia matrix, $\mathbf{C}(\boldsymbol{\eta}_b, \dot{\boldsymbol{\eta}}_b) = \mathbf{T}(\boldsymbol{\eta}_b)^T\mathbf{S}(\boldsymbol{\omega}_b)\mathbf{I}_b\mathbf{T}(\boldsymbol{\eta}_b) + \mathbf{T}(\boldsymbol{\eta}_b)^T\mathbf{I}_b\dot{\mathbf{T}}(\boldsymbol{\eta}_b) \in \mathbb{R}^{3\times3}$ is the Coriolis matrix, $\mathbf{S}(\cdot) \in \mathbb{R}^{3\times3}$ is the skew-symmetric operator, $g > 0$ is the gravity acceleration, $m > 0$ is the UAV mass, and $\mathbf{e}_3 = [0, 0, 1]^T \in \mathbb{R}^3$. The actuation part is dictated by the total thrust, $f_z > 0$, and the control torques around the axes of the body frame $\mathcal{B}$, collected in the vector $\boldsymbol{\tau}_b = [\tau_\phi, \tau_\theta, \tau_\psi]^T \in \mathbb{R}^3$.

III. METHODOLOGY

A. Problem statement

Consider a map of dimension $N_x \times N_y \in \mathbb{R}^2$ meters, where are placed $N_\xi > 0$ POIs, P_i, with $i = 1, \ldots, N_\xi$. These can correspond with checkpoints: areas where we want the drone to pass or chosen areas to discretize the map better (e.g., a bottleneck might be better described with more POIs than a large place). A given heat is associated with each POI. The problem is to minimize the sum of the heat values to guide the drone through the different POIs, considering the constraints of the map limits, the presence of obstacles, and the dynamic constraints of the system.

B. Optimisation problem

Let $\mathbf{d} = [d_1, ..., d_{N_d}]^T \in \mathbb{R}^{N_d}$ be the (measured) distances of the UAV from the (unknown) obstacles in the map along $N_d > 0$ chosen directions; $\boldsymbol{\xi} = [\xi_1, ..., \xi_{N_\xi}]^T \in \mathbb{R}^{N_\xi}$ be the vector collecting the heat values at the POIs; and $\boldsymbol{\iota} = [x, y, z, \phi, \theta, \psi, v_x, v_y, v_z, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T \in \mathbb{R}^{12}$ be the drone's state vector. Let $\boldsymbol{\zeta}(t) \in \mathbb{R}^N$, with $N = 12 + N_\xi + N_d$, be the state vector of the optimisation control problem and $\mathbf{u}(t) \in \mathbb{R}^4$ be the input vector. The state vector $\boldsymbol{\zeta} \in \mathbb{R}^N$ is defined as $\boldsymbol{\zeta} = [\boldsymbol{\iota}^T, \mathbf{d}^T, \boldsymbol{\xi}^T]^T$. The input vector is instead defined as $\mathbf{u} = [f_z, \tau_\phi, \tau_\theta, \tau_\psi]^T \in \mathbb{R}^4$. The dynamics of the UAV (1) is discretized with a sampling time $T_s > 0$ using the forward Euler method to obtain $\boldsymbol{\iota}_{k+1} = \mathbf{f}(\boldsymbol{\iota}_k, \mathbf{u}_k)$, where the subscript k is the discretized time variable. This discrete model is used as the prediction model of the NMPC. The prediction is made with a receding horizon composed of a specific number of steps into the future. This is denoted as the *prediction horizon* N_{opt}. The optimisation problem has the following structure

$$\begin{aligned} \min_{(\boldsymbol{\zeta}_k, \mathbf{u}_k)} \quad & \mathbf{J}(\boldsymbol{\zeta}_k, \mathbf{u}_k) \\ s.t. \quad & \\ & \boldsymbol{\zeta}(k_0) = \boldsymbol{\zeta}_0 \\ & \boldsymbol{\zeta}_{k+1} = [\,\boldsymbol{\iota}_{k+1}^T,\ \mathbf{d}_{k+1}^T,\ \boldsymbol{\xi}_{k+1}^T\,]^T \\ & \boldsymbol{\zeta} \in [\boldsymbol{\zeta}_{min}^T, \boldsymbol{\zeta}_{max}^T] \\ & \mathbf{u} \in [\mathbf{u}_{min}^T, \mathbf{u}_{max}^T] \end{aligned} \tag{2}$$

The linear quadratic cost function $\mathbf{J}(\boldsymbol{\zeta}_k, \mathbf{u}_k)$ is defined as

$$\begin{aligned} \mathbf{J}(\boldsymbol{\zeta}_k, \mathbf{u}_k) = \|\boldsymbol{\zeta}_{N_{opt},ref} - \boldsymbol{\zeta}_{N_{opt}}\|^2_{\mathbf{Q}_f} + \\ + \sum_{k=0}^{N_{opt}} \|\boldsymbol{\zeta}_{k,ref} - \boldsymbol{\zeta}_k\|^2_{\mathbf{Q}} + \|\mathbf{u}_{k,ref} - \mathbf{u}_k\|^2_{\mathbf{R}}, \end{aligned} \tag{3}$$

with the notation $\|\mathbf{x}\|^2_{\mathbf{A}} = \mathbf{x}^T\mathbf{A}\mathbf{x}$ to indicate the quadratic form and $\mathbf{Q}_f \in \mathbb{R}^{N\times N}$, $\mathbf{Q} \in \mathbb{R}^{N\times N}$, $\mathbf{R} \in \mathbb{R}^{4\times4}$ definite positive wights matrices. The notations $\boldsymbol{\zeta}_{k,ref} \in \mathbb{R}^N$ and $\mathbf{u}_{k,ref} \in \mathbb{R}^4$ represent the desired state and input vectors, respectively; while $\boldsymbol{\zeta}_{min} \in \mathbb{R}^N, \boldsymbol{\zeta}_{max} \in \mathbb{R}^N, \mathbf{u}_{min} \in \mathbb{R}^4$ and $\mathbf{u}_{max} \in \mathbb{R}^4$ represent the maximum and minimum limits for the states and inputs, respectively.

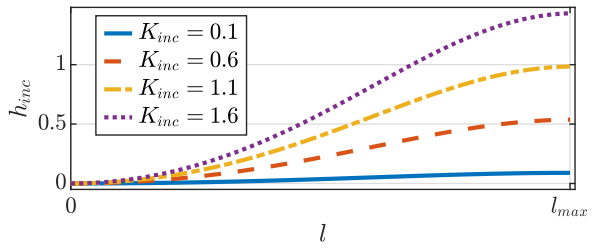


Fig. 2: Evolution of the incremental part h_{inc} for different K_{inc} gain values and $l_{max} = 10.0$ m

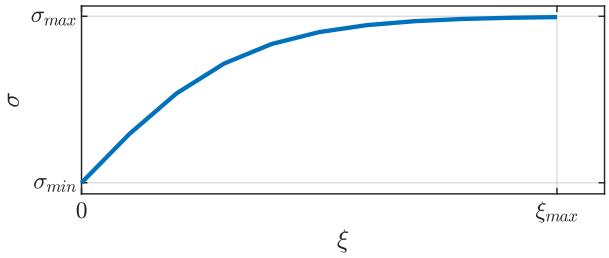


Fig. 3: Evolution of the i-th Gaussian distribution variance

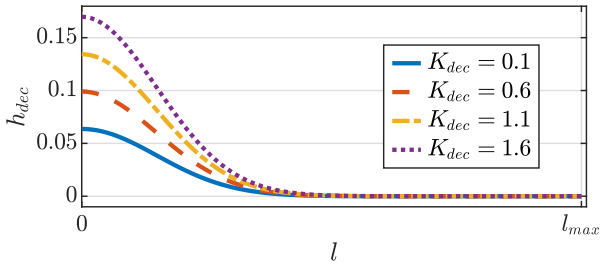


Fig. 4: Evolution of the descending part h_{dec} for different K_{dec} gain values, with $\xi_i = 0.8$, $\sigma_i = 1.5$ and $l_{max} = 10\,\text{m}$

C. Distance dynamics

The vector $\dot{\mathbf{d}} = [\dot{d}_1, ..., \dot{d}_{N_d}]^T \in \mathbb{R}^{N_d}$ is composed by the time derivative of the distances of the drone from the obstacles, towards the N_d directions. Considering an angle $\alpha_i \in \mathbb{R}$, with $i = 1, ..., N_d$, around the z axis of $\mathcal{B}$, the i-th distance derivative is given by

$$\dot{d}_i(\alpha_i, v_x, v_y) = -[\cos(\alpha_i)\ \sin(\alpha_i)] \begin{bmatrix} v_x \\ v_y \end{bmatrix}. \quad (4)$$

Equation (4) is discretized and employed within (2).

D. Heat dynamics

The vector $\dot{\boldsymbol{\xi}} = [\dot{\xi}_1, ..., \dot{\xi}_{N_\xi}]^T \in \mathbb{R}^{N_\xi}$ is composed by the derivative of the heat values relative to the POIs. Each element of the vector depends on the distance of the drone from the i-th point, according to

$$\dot{\xi}_i(l_i, \xi_i) = h_{inc}(l_i) + h_{dec}(l_i, \xi_i) \qquad i = 1, ..., N_\xi, \quad (5)$$

where $h_{inc}(l_i) \in \mathbb{R}$ is in charge of increasing the heat value and $h_{dec}(l_i, \xi_i) \in \mathbb{R}$ is the descending part. The incremental part has been constructed ad-hoc to achieve the behaviour shown in Figure 2, realised by

$$h_{inc}(l_i) = K_{inc_i}\ 2\ r_i^2\ l_i^2\ \text{sech}^2(r_i^2\ l_i^2), \quad (6)$$

where $\text{sech}(\cdot)$ is the hyperbolic secant function, $l_i(x, y, \mu_x, \mu_y) = \sqrt{(x - \mu_{x_i})^2 + (y - \mu_{y_i})^2}$ is the Euclidean distance between the drone and the i-th POI, $[\mu_{x_i}, \mu_{y_i}] \in \mathbb{R}^2$ are the coordinates of the i-th POI, $K_{inc_i} \in \mathbb{R}$ and $r_i \in \mathbb{R}$ are constants used to change the increasing rate. The maximum distance $l_{i_{max}}$ between the drone and the i-th POI depends on the POI's coordinates and the limits of the considered map. Then, by choosing $r_i = 1/l_{i_{max}}$, the incremental part reaches its maximum value when the drone is at the maximum distance from the POI. The descending part, instead, assumes greater value when the robot is near the POI and its evolution is dictated by

$$h_{dec}(l_i, \xi_i) = -(K_{dec_i} + \xi_i)\frac{1}{2\pi\sigma_i^2}\, e^{-\frac{l_i^2}{2\sigma_i^2}}, \quad (7)$$

where $K_{dec_i} \in \mathbb{R}$ is a constant that can change the decreasing rate, $\sigma_i = \sigma_{min} + \tanh(K_{\sigma_i}\,\xi_i)(\sigma_{max} - \sigma_{min})$ is the i-th Gaussian distribution variance with $\tanh(\cdot)$ being the hyperbolic tangent function, $K_{\sigma_i} \in \mathbb{R}$ is a constant gain, and $[\sigma_{min}, \sigma_{max}] \in \mathbb{R}^2$ are the bounds values. By choosing $K_{\sigma_i} = 3/\xi_{max}$, where $\xi_{max} \in \mathbb{R}$ is the heat value at which we want to have $\sigma_i = \sigma_{max}$, we obtain the evolutions of σ_i in Figure 3 and h_{dec} in Figure 4. The presented equations are firstly discretized and then employed in (2).

IV. Simulations

A. Description

Several simulations were performed to validate the proposed solution with different placements of the POIs, with and without obstacles on the map. Varying the POIs has helped understand the scouting behaviour and check if the drone gets stuck somewhere. The second block of simulations allowed seeing if the UAV could reach the destination despite obstacles.

The simulations have been performed on a standard personal computer with an Intel Core i7-10750H CPU and 16GB of RAM running Ubuntu 22.04 as the operating system. The code[1] has been developed in C++ and the library OCS2 [22] has been used to solve the optimal control problem along with the physics-engine-based simulator Gazebo and ROS as middleware. The flight dynamic of the drone has been simulated through the PX4 Autopilot [23] firmware using the software in the loop (SITL) simulation mode. The simulated drone is a standard commercial quadrotor of 55 cm motor-to-motor length, 1.282 kg of weight, and 400 g of payload [24].

The dimensions of the simulated area are $N_x = N_y = 10$ m, where five POIs have been placed. In order to deal with obstacles, $N_d = 4$ is considered, having four different angles α_i spaced $\pi/2$ apart from each other, obtaining

[1] https://github.com/prisma-lab/NMPC_UAV_repetitive_reconnaissance

the distances in the front, left, back, and right directions respect $\mathcal{B}$. The optimisation problem is implemented with a sampling time $Ts = 1$ ms and a prediction horizon $N_{opt} = 1500$. The constraints considered are: $x \in [0, 10]$ m; $y \in [0, 10]$ m; $z \in [1.5, 2.8]$ m; $\phi \in [-\frac{\pi}{6}, \frac{\pi}{6}]$ rad; $\theta \in [-\frac{\pi}{6}, \frac{\pi}{6}]$ rad; $\psi \in [-\pi, \pi]$ rad; $v_x \in [-1, 1]$ m/s; $v_y \in [-1, 1]$ m/s; $v_z \in [-0.5, 0.5]$ m/s; $\dot{\phi} \in [-2.0, 2.0]$ rad/s; $\dot{\theta} \in [-2.0, 2.0]$ rad/s; $\dot{\psi} \in [-2.0, 2.0]$ rad/s; $d_1, d_2, d_3, d_4 \in [0.40, \infty]$ m; $\xi_1, \xi_2, \xi_3, \xi_4 \in [0, 1]$; $f_z \in [0, 10]$ N; $\tau_\phi, \tau_\theta, \tau_\psi \in [-1.2, 1.2]$ N/m. Considering that we do not want to give a reference to the drone but minimise the overall heat value, the reference for the optimisation problem is a null vector $y_{ref} = \{\mathbf{0}\} \in \mathbb{R}^N$. For all the simulations, the following parameters were employed: $K_{inc_i} = 1.0$, $K_{dec_i} = 3.0$, $K_{\sigma_i} = 1.0$, $l_{i_{max}} = 10.0$, and $\xi_{i_{max}} = 1.0$. The gains found after a trial-and-error process are $\mathbf{Q} = diag([5, 5, 500, 5, 5, 100, 2, 2, 20, 10, 10, 10, 0, 0, 0, 0, 5000, 5000, 5000, 5000, 5000])$, $\mathbf{Q}_f = diag([2.5, 2.5, 100, 1, 1, 20, 0.5, 0.5, 20, 0.5, 0.5, 10, 0, 0, 0, 0, 0, 10, 10, 10, 10, 10])$ and $\mathbf{R} = diag([50, 500, 500, 500])$.

B. Case studies

1) Simulations with different points of interest: Different simulations were carried out with different POIs arrangements. The disposition of the POIs strongly influences the behaviour of the scouting drone and the time needed to inspect each point at least once. For instance, it has been noticed that if some POIs are placed in such a way as to create a conglomerate well separated from other POIs (see Fig. 5), the drone will often visit the area (i.e., a basin) delimited by the conglomerated POIs. However, this does not create a stuck condition: once the heat value of the POIs far from the basin reaches a value high enough to attract the drone, this last will escape from this region and visit the other POIs. Given the described configuration and all the parameters listed above, the UAV path resulting from the carried-out optimisation procedure is visible in Fig. 5. It is possible to observe a cyclic movement between the points (P_1, P_3, P_5). At the time $t \approx 114$ seconds, when the heat of P_2 is approaching the maximum value, the drone exits from the basin and navigates towards this point. With this behaviour, the cost function will be minimised once the drone reaches the point out of the basin because of the increasing heat value at P_2 (see Fig. 5c). Regardless of this cyclic solution, all the constraints in the NMPC problem are satisfied, and the drone can inspect each POI in less than 120 seconds. Faster scouting is obtained, considering a disposition where the POIs do not create an agglomerate (see Fig. 6). Figures 6b and 6c show that, in this case, the cost function reaches a minimum value in less than 45 seconds, and none of the probabilities assumes the maximum value. It is worth noticing that the drone does not need to pass at the POI's centre to reduce the heat because the heat has a spatial distribution determined by the variance σ. Moreover, depending on the value of the parameter K_{dec} (7), eventually, the drone will be repelled from the POI to satisfy the constraint $\xi_i > 0$.

2) Simulations with obstacles: A wall is first placed to create two isolated areas within the map. After, an obstacle is placed in the simulation environment to limit the movements of the overseer. In both simulations, intuitive results are obtained. In particular, in the simulation case with the wall (see Fig. 7), the drone flies through the points placed in the accessible part of the map and stops before the wall (see Fig. 8). This scenario obviously leads to the impossibility of minimizing the cost function, which will increase because of the heat of the POIs placed in the inaccessible part of the map (see Fig. 9). Instead, the simulation with the obstacle showed that the drone can avoid collisions and visit all the POIs (see Fig. 10). Obviously, the task is longer than the simulation in free space ($\approx$ 90 seconds compared to $\approx$ 45 seconds). Regardless of this delay, the problem has been solved, satisfying all the constraints and minimising the cost function (see Fig. 11).

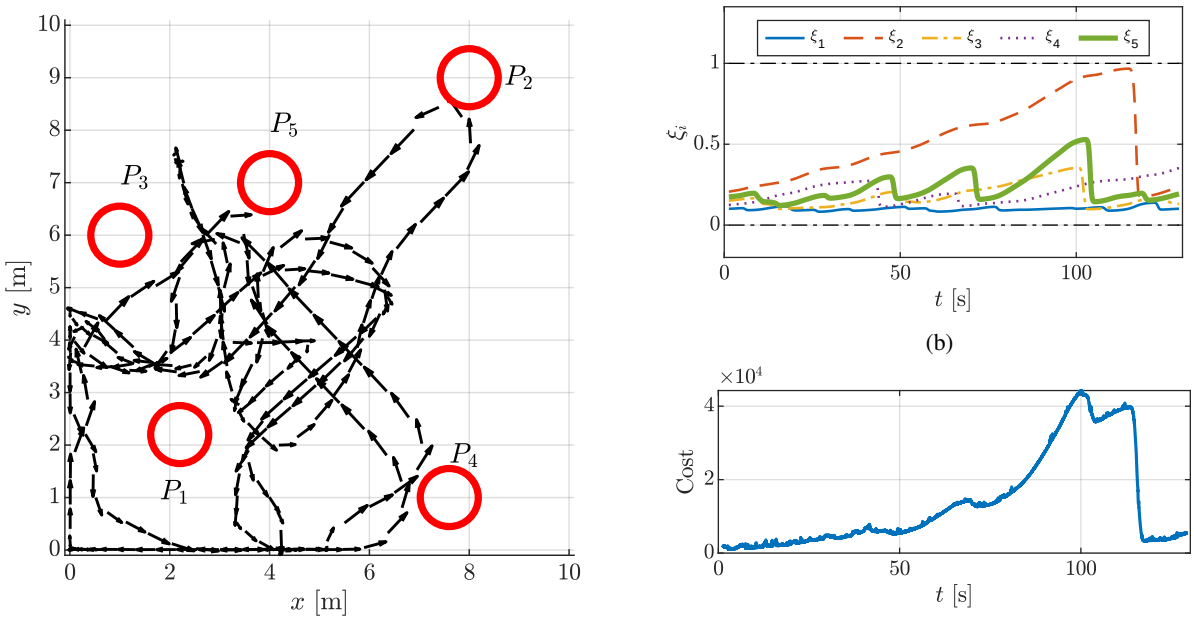
V. Experiments

A. Description

Experiments on a real multirotor platform in a flying arena (see Fig. 12a) were conducted to validate the proposed approach further. Due to the reduced dimensions of the arena, only 3 POIs were placed. The multirotor UAV is a coaxial octocopter with a 30×30 cm frame which has on top a Pixhawk Cube Orange flight controller that runs the last release of PX4 v1.13.0. Moreover, the drone has an Intel UP^2 as the onboard computer, running the Ubuntu 18.04 operative system, ROS 1 as middleware, and the proposed NMPC. The flying arena measures $N_x = 3$ m and $N_y = 5$ m. The UAV position has been measured with the motion capture system Optitrack [25] using fifteen high-speed IR cameras and sent to the drone using the Mavros ROS package through Wi-Fi. For this experiment, there are no obstacles inside the flight area because the used drone is not equipped with distance sensors.

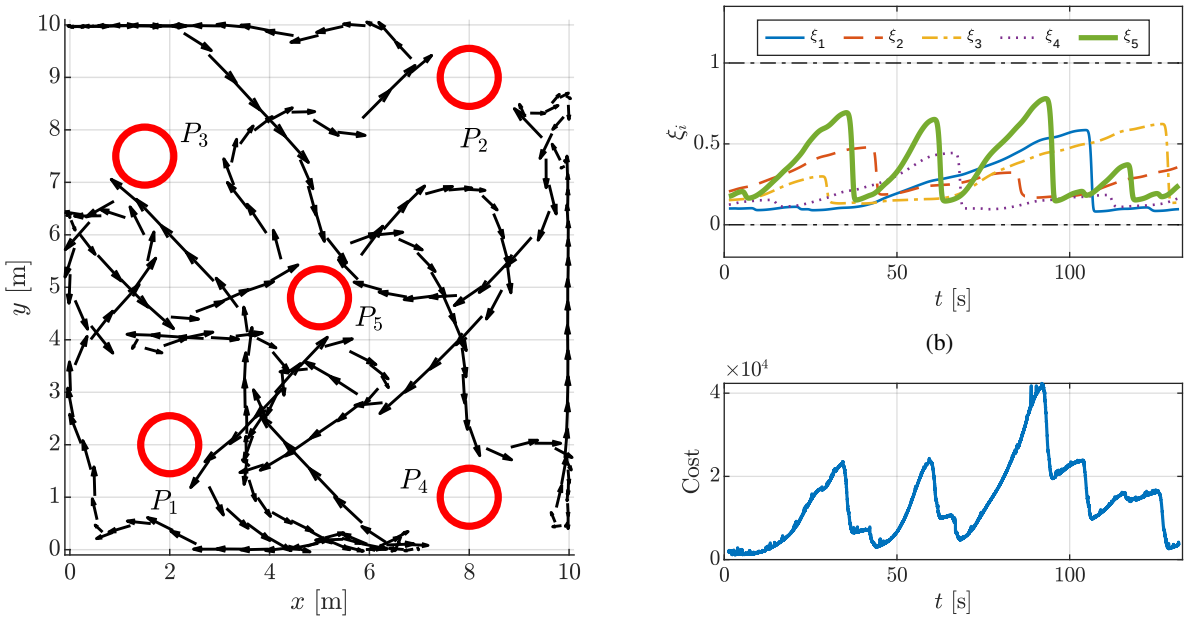
B. Case studies

The three POIs have different values for the incremental rates, namely, $K_{inc_1} = 1.0$, $K_{inc_2} = 1.5$, $K_{inc_3} = 0.5$. For greater safety, the maximum linear velocities have been decreased to $x, y \in [-0.6, 0.6]$ m/s, while the rest of the parameters have been left unchanged. The obtained results are reported in Fig.s 12b-12d. From the experimental tests, it is possible to infer that the NMPC satisfies the dynamic constraints of the real multirotor and the map boundaries, generating smooth trajectories guiding the UAV through the POIs. For space reasons, it has not been possible to show in this paper that all the constraints within the NMPC are



(a) (b) (c)

Fig. 5: Results of the simulation with POIs forming a conglomerate: in (a) is shown the path followed by the drone during the simulation, in (c) is shown the cost evolution, in (b) is shown the evolution of the heat values



(a) (b) (c)

Fig. 6: Results of the simulation with POIs more spatially distributed: in (a) is shown the path followed by the drone during the simulation, in (c) is shown the cost evolution, in (b) is shown the evolution of the heat values

satisfied: this is indeed shown in the video[2] that comes with this article.

[2]https://youtu.be/IM8WvibB6Zo

VI. Discussion

The proposed framework considers the system's dynamic that performs the surveillance task and plans the trajectory online through the different POIs, considering their heat value. This value changes over time and can have a desired

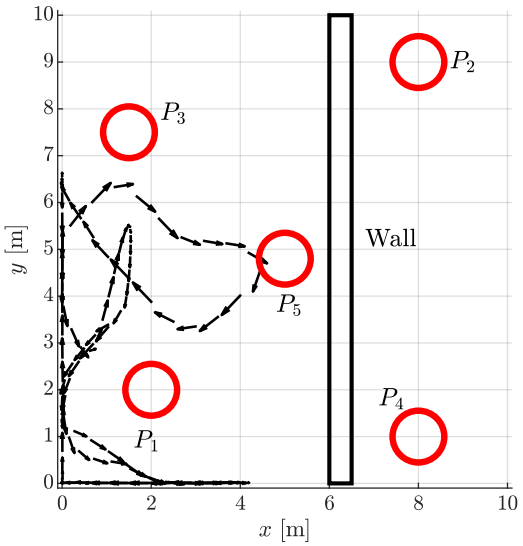


Fig. 7: UAV trajectory in the map with a wall. The circles represent the points of interest and the arrows represent the path of the UAV.

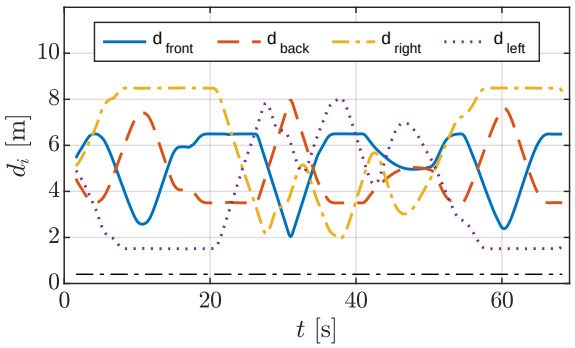


Fig. 8: UAV distances while navigating in the map with a wall.

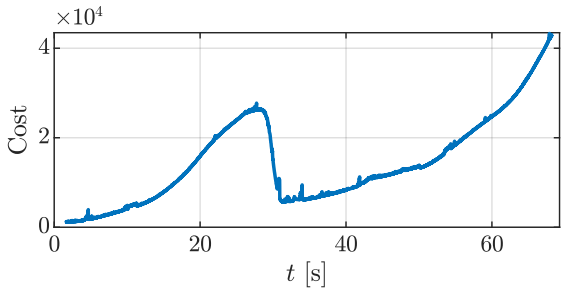


Fig. 9: Cost function for the navigation in the map with a wall. Notice that, due to the POI behind the wall, the cost function cannot be minimised.

evolution law. In that way, the overseer will not follow a pre-planned static path, resulting in less predictable behaviours. Unlike an offline approach, this framework does not require the model of the environment structure. Instead, the limits

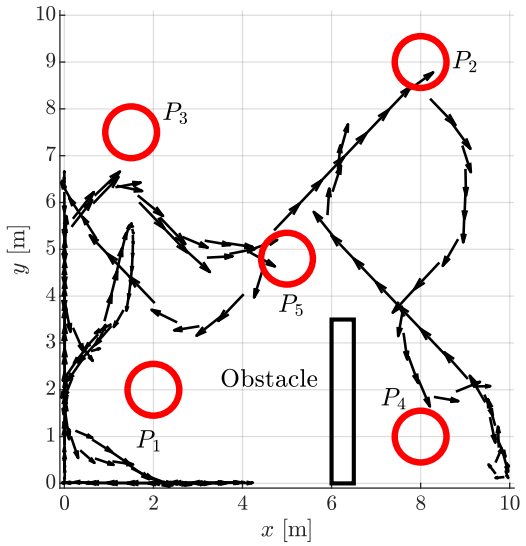


Fig. 10: UAV trajectory in the map with an obstacle. The circles represent the POIs and the arrows represent the path of the UAV.

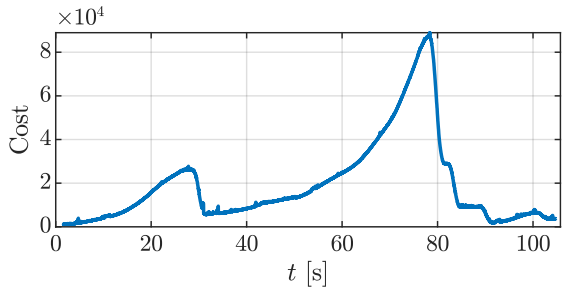


Fig. 11: Cost function for the navigation in the map with an obstacle.

given by the environment are considered thanks to the measurement of the distance from the obstacles. This approach simplifies the structure of the problem but also limits the performance to avoid obstacles. In fact, the overseer will avoid collisions with the obstacles by measuring the distance. However, due to the lack of knowledge of the obstacles' geometry, it is impossible to plan a suitable trajectory to circumnavigate them.

Besides, in this paper, the proposed framework has been applied with less than 10 POIs and a multirotor as a scouting system, using the dynamic model with thrust and torques inputs. Considering that the solving time of the NMPC highly depends on the number of state variables and their evolution law, the system dynamics can be simplified (e.g., using the kinematic model only) if the area to be recognized requires a higher number of POIs. Moreover, the parameters of the equations can be tuned depending on the map's dimensions and the position arrangement of the POIs. Indeed, for example, if there is a POI far away from the others, it must be taken into account and give it a slow heat's increasing dynamic.

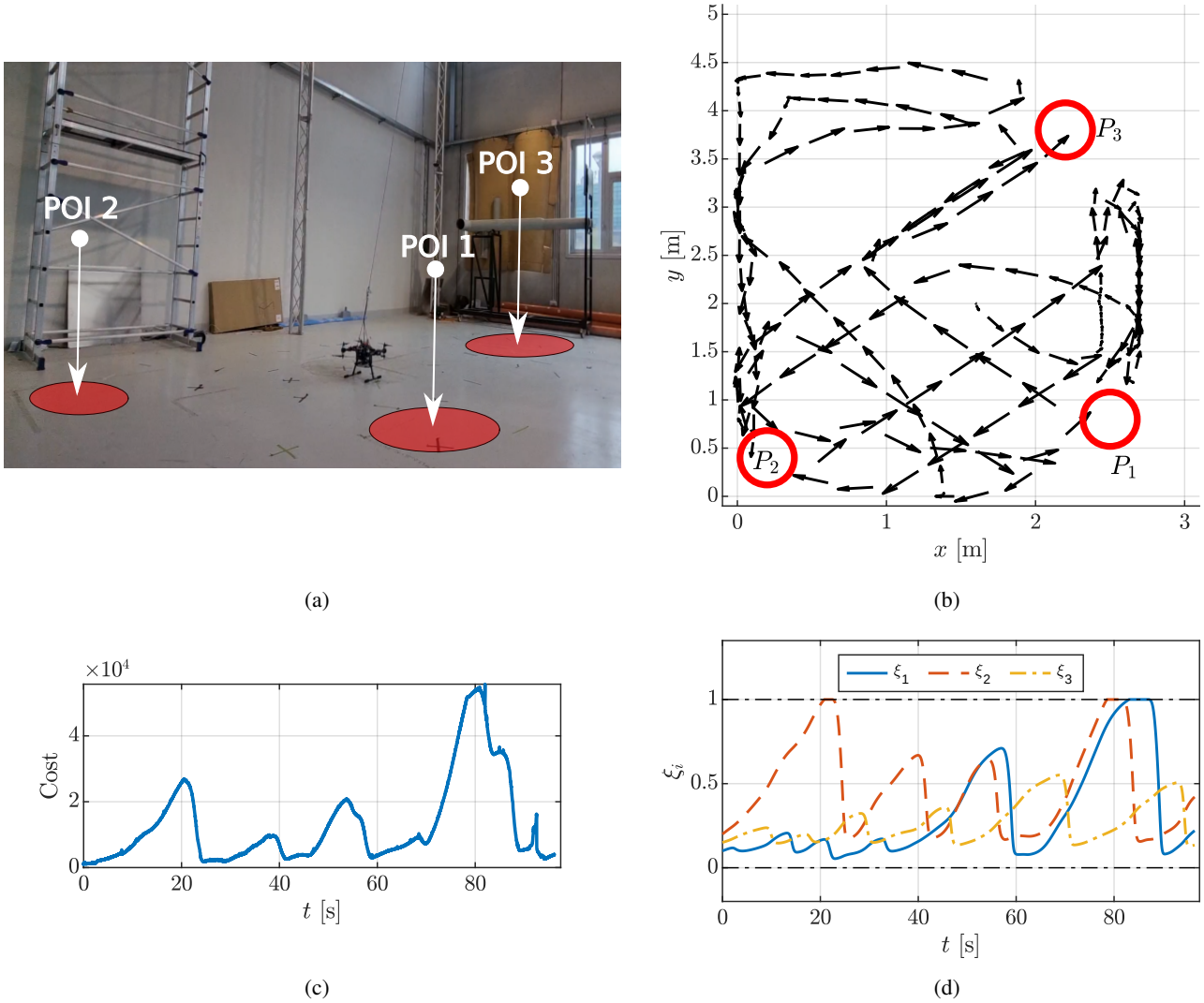


Fig. 12: Results of the experimental test in the flying arena: in (a) is shown the UAV and the flight arena with the POIs, in (b) is shown the path followed by the drone during the experiment, in (c) is shown the cost evolution, in (d) is shown the evolution of the heat values. The video is available at `https://youtu.be/IM8WvibB6Zo`

VII. Conclusion and Future Work

This paper proposed a novel solution for a repetitive reconnaissance problem. In this type of problem, the main goal is to perform a reconnaissance of a designed area, crossing multiple times some areas/points of interest. The proposed solution consists of a general framework using an NMPC architecture with a UAV, including collision avoidance with environmental obstacles and constraints given by the UAV dynamics. Simulations and experimental results show that the overseer is successfully guided through the obstacles toward the pre-defined POIs. It has also been experimentally verified that the drone does not get stuck in any particular area, always moving toward any POI to minimize the cost function. Nevertheless, the proposed solution exhibits some limitations: (i) the solution strongly depends on the tuned gains and the parameters; (ii) the system cannot predict a path to avoid obstacles but only to avoid collisions.

This work can be further extended. First, it is possible to give different increment rates to the probability so that the drone can visit some POIs more than others. Then, other constraints can be added to the NMPC to speed up the recognition without letting the agent stay still in a position for a few seconds. Moreover, adding a negative Gaussian distribution in the map might also represent a *no-fly zone* where the agents must not pass through. Another extension might be the possibility of addressing a multi-agent system. Indeed, the overall proposed framework can be used as a baseline for a multi-robot or multi-agent approach, where a flock of drones must collaborate or share the knowledge with some security cameras to survey a specific area and where the evolution of the probability can consider each agent's contribution.

REFERENCES

[1] M. Idrissi, M. Salami, and F. Annaz, "A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms," *Journal of Intelligent & Robotic Systems*, vol. 104, no. 2, p. 22, 2022.

[2] A. Suarez, J. Cacace, and M. Orsag, "Aerial Robotics for Inspection and Maintenance: Special Issue Editorial," *Applied Science*, vol. 12, no. 7, 2022.

[3] S. Kulkarni, V. Chaphekar, M. M. U. Chowdhury, F. Erden, and I. Guvenc, "UAV Aided Search and Rescue Operation Using Reinforcement Learning," *2020 SoutheastCon*, vol. 2, pp. 1–8, 2020.

[4] P. Panagiotou, D. Mitridis, T. Dimopoulos, S. Kapsalis, S. Dimitriou, and K. Yakinthos, "Aerodynamic design of a tactical Blended-Wing-Body UAV for the aerial delivery of cargo and lifesaving supplies," *AIAA Scitech 2020 Forum*, 2020.

[5] S. S. Mansouri, C. Kanellakis, E. Fresk, B. Lindqvist, D. Kominiak, A. Koval, P. Sopasakis, and G. Nikolakopoulos, "Subterranean MAV navigation based on nonlinear MPC with collision avoidance constraints," ser. IFAC-PapersOnLine, vol. 53. Elsevier B.V., 2020, pp. 9650–9657.

[6] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for Collision Avoidance and Control of UAVs With Dynamic Obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6001–6008, 2020.

[7] T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira, "Survey on Coverage Path Planning with Unmanned Aerial Vehicles," *Drones*, vol. 3, no. 1, p. 4, 2019. [Online]. Available: www.mdpi.com/journal/drones

[8] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of Multiple UAVs for Persistent Surveillance: Algorithm and Flight Test Results," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1236–1251, 2012.

[9] D. Fehr, L. Fiore, and N. Papanikolopoulos, "Issues and solutions in surveillance camera placement," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3780–3785.

[10] P. Fazli, A. Davoodi, and A. K. Mackworth, "Multi-robot repeated area coverage: Performance optimization under various visual ranges," in *2012 Ninth Conference on Computer and Robot Vision*, 2012, pp. 298–305.

[11] N. Dilshad, J. Hwang, J. Song, and N. Sung, "Applications and challenges in video surveillance via drone: A brief survey," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 728–732.

[12] J. Araújo, P. Sujit, and J. Sousa, "Multiple UAV Area Decomposition and Coverage," *2013 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pp. 30–37, 2013.

[13] M. A. Luna, M. S. A. Isaac, A. R. Ragab, P. Campoy, P. F. Peña, and M. Molina, "Fast Multi-UAV Path Planning for Optimal Area Coverage in Aerial Sensing Applications," *Sensors (Basel, Switzerland)*, vol. 22, no. 6, p. 2297, 2022.

[14] C. Ercolani, L. Tang, A. Humne, and A. Martinoli, "Clustering and informative path planning for 3D gas distribution mapping: Algorithms and performance evaluation," in *2022 IEEE International Conference on Robotics and Automation*, 2022.

[15] K. Jakkala and S. Akella, "Probabilistic gas leak rate estimation using submodular function maximization using routing constraints," in *2022 IEEE International Conference on Robotics and Automation*, 2022.

[16] J. Scherer and B. Rinner, "Persistent multi-uav surveillance with energy and communication constraints," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 1225–1230.

[17] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2, 2009, pp. 82–85.

[18] K. S. Lee, M. Ovinis, T. Nagarajan, R. Seulin, and O. Morel, "Autonomous patrol and surveillance system using unmanned aerial vehicles," in *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, 2015, pp. 1291–1297.

[19] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics a survey," vol. 31, pp. 299–316, 2011.

[20] B. P. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *The International Journal of Robotics Research*, vol. 25, no. 4, pp. 299–315, 2006. [Online]. Available: https://doi.org/10.1177/0278364906065023

[21] B. S. A. Ollero, *Aerial robotic manipulation*. Springer, 2019.

[22] Ocs2 1.0.0 documentation. [Online]. Available: https://leggedrobotics.github.io/ocs2/

[23] Px4 autopilot. [Online]. Available: https://px4.io/

[24] Iris - the ready to fly uav quadcopter. [Online]. Available: http://www.arducopter.co.uk/iris-quadcopter-uav.html

[25] Optitrack website. [Online]. Available: https://optitrack.com/