POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Autonomous navigation and obstacle avoidance for interconnected tethered drones in a partially unknown environment

Laurea Magistrale in Automation and Control Engineering - Ingegneria dell'Automazione

Author: Fabrizio Cirillo

Advisor: Prof. Lorenzo Fagiano

Co-advisor: Danilo Saccani, Michele Bolognini

Academic year: 2021-2022

1. Introduction

The cost of Unmanned Aerial Vehicles (UAVs) in the last years has decreased. As a consequence the interest in such vehicles has increased as well, most of all in some field of industry and academia. They are used in many applications such as mapping [2], monitoring of building zone [5] and some other due to their versatility. Nonetheless they have the drawback of having a low flight time, in fact for time consuming applications a system of tethered drones is generally used. The topic of this thesis work consists in a high level controller used by a system, which is called STEM (System of TEthered Multicopters) [3]. STEM is composed by a chain of multicopter drone, one tethered to each other with the tail of the chain tethered to a ground station. The tethers permits communication and power supply to the drones [1].

2. Main Contributions

In this thesis works, the environment is partially known through the usage of a map. This knowledge is provided to an offline planner which generates the target points for the drones. The online path following algorithm is accomplished by MPC (Model Predictive Control): taking as inputs the map and the readings of LiDAR sensors it produces a reference position for the drones at each sampling time. Once the assigned targets are reached, a second configuration is found by offline planner. Finally, the second set of targets are reached, maybe including some rewind process. To summarize, the main contributions of this research are:

- the formulation of a first optimal offline planner operating in a well known environment whose aim is to compute an optimal configuration which the drones have to track;
- the development of a path following algorithm, based on MPC, which aims to bring the system to the desired configuration respecting some linear and non-linear constraints. They are related to obstacle avoidance, distance between drones, position, velocity and acceleration;
- the formulation of a second optimal offline planner which finds a new optimal configuration taking as input the final position reached by the drones and a new target as-

signed to the leader;
- the development of an online strategy which chooses the best strategy concerning the drones, selecting between the drone rewound or directly assigning the new targets in a precise order;
- the development of a rewind path planner which is capable of rewinding the drones according to the chosen aforementioned strategy;
- test of the approach in simulation using a simplified model of the drones called "oriented control model".

3. Model of STEM

Nonlinear model of STEM is depicted in [3]. Its linear approximation is used for the high-level controller.

3.1. Model of the drones

The model of the quadcopter is the typical one reported in literature [4], with contribution of forces and moments given by the tethers attached to the drone. To simplify the treatment all drones are assumed to be identical. The equation of a single drone is the following:

$$m_{d,i}(t) = \overline{m}_{d,i} + m_{w,i}(t) + \frac{1}{2}m_{t,i}(t), \qquad (1)$$

where $\overline{m}_{d,i}$ is the mass of the vehicle alone, $m_{w,i}$ is the weight of the winch and the stored cable and $m_{t,i}$ is the weight of the extended cable which connects drone i to drone $i+1$. The inputs of the model, lift force and a drag torque, one for each rotor, are:

$$\begin{aligned} L_{i,j}(t) &= b\Omega_{i,j}^2, \quad j = 1, ..., 4 \\ T_{i,j}(t) &= d\Omega_{i,j}^2, \quad j = 1, ..., 4, \end{aligned} \qquad (2)$$

where b, d are respectively the lift and drag coefficient. They are subsequently recombined in a linear combination for control purpose and the model, which is not reported here for the sake of brevity, is obtained.

3.2. Model of Tether and Winch

The number of drones composing the chain is assumed to be N_d, where $i = N_d$ correspond to the drone at the end of the string. Winches are identified by the progressive index i, where $i = 0$ corresponds to the ground station, and the subsequent $i = 1, ..., N_d - 1$ correspond with the index used for the drones. Even the cable is identified with the index of the corresponding winch. The angular position and velocity of the i-th winch, $\theta_i(t), \dot{\theta}_i(t)$, are its state. It is assumed that when the measured position $\theta_i(t) = 0$ the cable is completely wound around the winch. Then, assuming that the whole cable can be coiled on a single layer, i.e. the external radius of the winch is independent with respect the length of unreeled tether, it is possible to compute the mass of the winch:

$$m_{w,i}(t) = \overline{m}_{w,i} + \left(\overline{l}_i - r_{e,i}\theta_i(t)\right)\rho_{t,i}, \qquad (3)$$

where $r_{e,i}$ is the external radius, $\rho_{t,i}$ is the unitary mass of the tether per length, $\overline{l}_i$ is the overall length of the tether i and $\overline{m}_{w,i}$ is the mass of the winch with no tether wounded. The moment of inertia of the winch, which can be approximated as a hollow drum, with internal radius $r_{\mathrm{i},i}$ is computed as:

$$J_{w,i}(t) = \frac{1}{2}m_{w,i}(t)\left(r_{e,i}^2 + r_{\mathrm{i},i}^2\right) \qquad (4)$$

. The winch is physically defined by a viscous friction coefficient, which is assumed to constant and is denoted with $B_{w,i}$. The winch torque, which is a control input, is denoted as $u_{w,i}$ and is bounded in the interval $[\underline{u}_{w,i}, \overline{u}_{w,i}]$. Then, the elongation of the tether $e_{t,i}(t)$ is computed as:

$$e_{t,i}(t) = max\left(0, \left\|\boldsymbol{p}_{i+1}^g(t) - \boldsymbol{p}_i^g(t)\right\|_2 - r_{e,i}\theta_i(t)\right) \qquad (5)$$

Finally, the vector of forces which the tether exerts on the drone, expressed in global coordinates, is calculated from the elongation $e_{t,i}$ of the tether itself and its stiffness K_t, which is assumed to be constant:

$$\boldsymbol{F}_{t,i}^g(t) = K_{t,i}e_{t,i}(t)\frac{\boldsymbol{p}_{i+1}^g(t) - \boldsymbol{p}_i^g(t)}{\left\|\boldsymbol{p}_{i+1}^g(t) - \boldsymbol{p}_i^g(t)\right\|_2}, \qquad (6)$$

where $\boldsymbol{p}_0$ is the ground station position. Therefore, using again Newton's law, it is possible to derive the state equation of the winch, recalling to the equilibrium of moments around the axis of rotation:

$$\ddot{\theta}_i(t) = \frac{1}{J_{w,i}(t)}\left(r_{e,i}\left\|\boldsymbol{F}_{t,i}^g\right\|_2 - \beta_{w,i}\dot{\theta}_i(t) + u_{w,i}(t)\right) \qquad (7)$$

Aerodynamic drag, as assumption, is neglected, considering negligible the speed of wind relative to the tether. In (6), it is assumed that drones i and $i+1$ exchange forces along the direction which connects their centre of mass, not along the direction of the two points where the cable is connected to the vehicles. Since the distance between the drones is higher than the distance between the center of mass and the cable attachment, this assumption can be considered valid. The complete nonlinear model is the same obtained in [3].

3.3. Control Oriented Model

STEM is assumed to be controlled by a low level controller with high working frequency as done in [3]. The high level planner aims to produce the references $\boldsymbol{P}_{ref} = \begin{bmatrix} P^x_{ref} & P^y_{ref} & P^z_{ref} \end{bmatrix}^T$ for the previous mentioned controller. The model used here is the so called "control oriented" model and is a 2D LTI model:

$$\begin{bmatrix} \dot{\boldsymbol{P}}(t) \\ \dot{\boldsymbol{V}}(t) \end{bmatrix} = \begin{bmatrix} 0^{2\times2} & I^{2\times2} \\ -K_1 & -K_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}(t) \\ \boldsymbol{V}(t) \end{bmatrix} + \begin{bmatrix} 0^{2\times2} \\ K_1 \end{bmatrix} \boldsymbol{P}_{ref}(t), \tag{8}$$

where $\boldsymbol{V}(t) = \dot{\boldsymbol{P}} \in \mathbb{R}^2$, $\boldsymbol{P}(t) \in \mathbb{R}^2$ are drone velocities and positions, while $0^{2\times2}$, $I^{2\times2}$ are zero and identity matrices. The acceleration of the drone can be computed as:

$$\boldsymbol{A}(t) = K_1 \left(\boldsymbol{P}_{ref}(t) - \boldsymbol{P}(t) \right) + K_2 \boldsymbol{V}(t), \tag{9}$$

where K_1, K_2 are gains tuned after a procedure of closed loop identification. Then, it is discretized with zero order hold (ZOH) method using $T_s = 0.5s$, which is a suitable value given the high level nature of the navigation control system:

$$\boldsymbol{x}_i(k+1) = A_i \boldsymbol{x}_i(k) + B_i \boldsymbol{u}_i(k), \tag{10}$$

where $\boldsymbol{x}_i(k) = \begin{bmatrix} \boldsymbol{P}_i(k) & \boldsymbol{V}_i(k) \end{bmatrix}^T \in \mathbb{R}^4$ is the state of vehicle and $\boldsymbol{u}_i(k) = \boldsymbol{P}_{ref}(k)$ correspond to its input. To conclude, the full control oriented model composed by N_d drones is written as:

$$\boldsymbol{x}(k+1) = A_t \boldsymbol{x}(k) + B_t \boldsymbol{u}(k), \tag{11}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^{4N_d}$ is the vector of the full states, $\boldsymbol{u}(k) \in \mathbb{R}^{2N_d}$ is the vector of inputs, $A_t = diag(A_1, \dots, A_{N_d}$ and $B_t = diag(B_1, \dots, B_{N_d})$ are state and input matrices. In conclusion, it is introduced the selection matrix $\mathbb{P}_{xy} \in \mathbb{R}^{4N_d \times 2N_d}$ which has the following property: $\boldsymbol{x}(k)\mathbb{P}_{xy} = \left[\boldsymbol{x}_{1_{1:2}}(k)^T, \dots, \boldsymbol{x}_{N_{d_{1:2}}}(k)^T \right]^T$. It selects (x, y) components.

3.4. Sensor

Each drone is provided with an IMU (Inertial Measurement Unit), GPS (Global Positioning System), and LiDAR sensor (Light Detection And Ranging). This last sensor is used to perceive the environment and permits the drones to localize obstacles during the flight. Each sensor produces a vector $\gamma(k)$ of $N_r = \frac{2\pi}{\alpha_s}$ measurements, where α_s corresponds to the angular resolution. The aforementioned measurements can be also expressed as a distance through the vector $d_i(k) = \gamma_i \left[cos(\vartheta_i) \quad sin(\vartheta_i) \right]$, $i = 1, \dots, N_r - 1$, $d(k) \in \mathbb{R}^{2\times N_r - 1}$ where d_i represents the distance from the sensor to the closest obstacle in the direction expressed by ϑ_i. If in that direction no obstacle is detected, the value assigned to γ_i is equal to R_L, the maximum range which the sensor can measure. Since the measurements from LiDAR sensors are provided with a frequency of at least 10Hz, it is possible to directly implement them in the algorithm in subsection (4.2).

3.5. Environment

The considered environment is a 2D one. To simplify the formulation of constraints in the offline planner, the obstacles are considered to be ellipses. In fact, a compact set can describe them, involving the shape matrix H_j in this way:

$$O_j := \left\{ \chi \in \mathbb{R}^2 : (\chi - \chi_{c_j})^T H_j (\chi - \chi_{c_j}) \leq 1 \right\}, \tag{12}$$

where H_j, χ_{c_j} represents respectively the geometric shape and the coordinate of the center of the j-th ellipse. Moreover, it is possible to define the set of all obstacles as $O := \bigcup_{j=1}^{N_0} O_j$, with N_0 the total number of obstacles. Furthermore, ellipses are used also because is always possible to approximate a set of non-convex obstacles as a set of ellipses.

4. Proposed approach

First, offline mission planner is depicted, its aim is to find a obstacle free optimal configuration for the drones starting from the target assigned

to the leader one. Secondly, a Linear Program (LP), used to find the optimal maximum ellipse inside merged readings is described. After that, an online path following algorithm which bring the drones to the previously found optimal configuration, is presented. Subsequently, a new optimal configuration for the drones is obtained assigning a new target to the leader drone as well. Then, a strategy algorithm selects the right strategy to impose to the system, choosing between a backtrack strategy or a new path planning one. In conclusion, the drones are again brought towards their final configuration.

4.1. Offline Path Planner

It is useful to introduce the concept of configuration of the system, it is a vector gathering the position of all drones: $\boldsymbol{C}(t) = \bigcup_{i=1}^{N_d} \boldsymbol{P}_i(t)$. Such configuration is said to be admissible in the environment if the positions of all drones and the tethers connecting them belong to the free space $\boldsymbol{S}_{free} \coloneqq \mathbb{R}^2 \setminus O$. This Optimization Program (OP) finds a optimal configuration $\boldsymbol{C}^*$ for the drones receiving as input $\boldsymbol{P}_T$, the target of the leader drone. The optimisation variables are the position $\boldsymbol{P}_i$ of the drones. The optimal configuration computed by the offline path planner is $\boldsymbol{C}^*$. The distance between the leader and the aforementioned target and the distance between consecutive drones are minimized using this cost function:

$$J = \alpha \left\| \boldsymbol{P}_T - \boldsymbol{P}_{N_d} \right\|_2^2 + \beta \sum_{i=0}^{N_d-1} \left\| P_{N_d-i} - P_{N_d-i-1} \right\|_2^2, \tag{13}$$

where α and β are two weights. The nonlinear inequality constraints are related to the maximum and minimum length $\left(\bar{l}, \underline{l}\right)$ of the tether between consecutive drones:

$$\underline{l} \leq \|\boldsymbol{P}_{i+1} - \boldsymbol{P}_i\|_2^2 \leq \bar{l}, \quad i = 1, \ldots, N_d - 1, \tag{14}$$

and to the distance between $\boldsymbol{P}_i$ and obstacles. In addition, the distance between obstacle and tether must be at least σ (user-defined variable). These last constraints are obtained using the perpendicular line method (Fig 1).

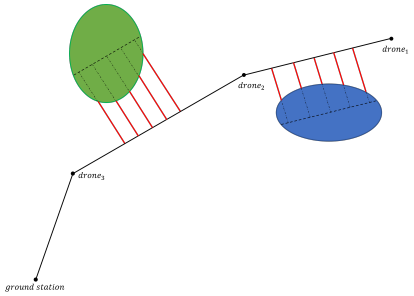

Figure 1: A representation of perpendicular line method approach with ellipse

The parallel to the tether passing through the center of ellipse is traced, consequently discretized points on this line are found. Each distance between ellipse and tether is calculated and then is imposed to be higher than σ.

4.2. Convex Approximation of Free Space

The set containing all the LiDAR measurements of the i-th drone at time k is defined as $L_i(k) \coloneqq \{d_0(k), \ldots, d_{N_r-1}(k)\} \in \mathbb{R}^2$. To calculate the non-convex area defined by the LiDAR readings of two consecutive drones i and $i+1$, a set of overlapping measurements is defined as:

$$\begin{aligned} L_d(k) = \{ & d_m(k) \in L_i(k),\ d_n(k) \in L_{i+1}(k) : \\ & \|d_m(k) - \boldsymbol{P}_{i+1}(k)\|_2 < R_L \wedge \\ & \|d_m(k) - \boldsymbol{P}_i(k)\|_2 = R_L, \\ & \|d_n(k) - \boldsymbol{P}_i(k)\|_2 < R_L \wedge \\ & \|d_n(k) - \boldsymbol{P}_{i+1}(k)\|_2 = R_L, \\ & \forall m, n = 0, \ldots, N_r - 1.\} \end{aligned} \tag{15}$$

Merged readings are selected as:

$$L_{i,i+1}(k) = L_i(k) \cup L_{i+1}(k) \setminus L_d(k) \tag{16}$$

The cardinality of $L_{i,i+1}(k)$ is denoted as N_{mr}. Then, a maximum ellipse is found in this merged readings through a Linear Program (LP), subsequently, a polytope is derived starting from the ellipse and it is expanded maximizing its area. Considering the equation of an ellipse in Cartesian plane

$$k_1 \tilde{x}(x, y)^2 + k_2 \tilde{y}(x, y)^2 = 1 \tag{17}$$

and defining the functions $\tilde{x}, \tilde{y} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$:

$$\begin{aligned} \tilde{x}(x, y) &= (x - x_0)\cos(\alpha) + (y - y_0)\sin(\alpha) \\ \tilde{y}(x, y) &= (y - y_0)\cos(\alpha) - (x - x_0)\sin(\alpha) \end{aligned} \tag{18}$$

where (x_0, y_0) correspond to the coordinate of the center of the ellipse and α is the angle of rotation of its major axis, it is possible to define the following LP (19):

$$\begin{gathered}\min_{k1,k2} k_1 + k_2 \\ s.t. \\ k_1\tilde{x}(P_{i_1}, P_{i_2}) + k_2\tilde{y}(P_{i_1}, P_{i_2}) \leq 1, \\ k_1\tilde{x}(P_{i+1_1}, P_{i+1_2}) + k_2\tilde{y}(P_{i+1_1}, P_{i+1_2}) \leq 1 \\ k_1\tilde{x}(d_m) + k_2\tilde{y}(d_m) \geq 1, \ \ \forall m = 0, \dots, N_{mr} - 1 \\ \boldsymbol{Ak} \leq \boldsymbol{b},\end{gathered}$$

where the first two constraints regard the inclusion of drone i and $i+1$ positions in the ellipse, the third concerns the LiDAR readings which have to lie outside the ellipse and the final one is used to impose some behaviour (i.e. semi-major axis higher than semi-minor one). The base polytope is computed through discretization of the ellipse, then, it is expanded to have the maximum under-approximation of the free space $\boldsymbol{D}_j$, $j = 1, \dots, N_d$. Finally, the polytope in such way obtained is reduced to have an offset between it and the readings. For each pair of drones the polytopes are collected in the set:

$$S(k) = \{D_j(k), \ \ j = 1, \dots, N_d\} \tag{20}$$

4.3. Online Path Following

A real-time path following strategy is now necessary to bring the drones to the optimal configuration $\boldsymbol{C}^*$ obtained in subsection (4.1). LiDAR readings are used to derive an approach to react to unexpected obstacles. The trajectory simulated over the horizon $N \in \mathbb{N}$ is considered safe because the drones stop inside the region $S(k)$ which is obstacle-free. MPC algorithm consists on solving at each iteration time a Finite Horizon Optimal Control Problem (FHOCP), obtaining $\boldsymbol{U}^*$. Then, only the first vector of optimisation variables $\boldsymbol{u}^*(0)$ is applied. The cost function of this OP regard the minimisation of the square distance between state goals, collected in $\boldsymbol{x}_g(k) = \left[\boldsymbol{x}_{g_i}(k), \dots, \boldsymbol{x}_{g_{N_d}}(k)\right]^T \in \mathbb{R}^{4N_d}$ and actual state $\boldsymbol{x}(k)$ and and the minimization of two consecutive inputs. In fact, we can write J=J_1+J_2 where

$$J_1 = \sum_{j=1}^{N} \sum_{i=1}^{N_d} \left\| Q(\boldsymbol{x}_{i_{(1:2)}}(j|k) - \boldsymbol{x}_{g_i}(k)) \right\|_2^2 \tag{21}$$

and

$$J_2 = \sum_{j=1}^{N} \| T_u(\boldsymbol{u}(j|k) - \boldsymbol{u}(j-1|k)) \|_2^2 \tag{22}$$

where $Q \in \mathbb{R}^{3\times3}$, $T_u \in \mathbb{R}^{6\times6}$ are positive-definite weighting matrices. The vector of the goals is computed via a path following approach. The path is the set of all sampled points on the lines connecting all the target of the drones, where all the points on it have a predefined distance one to each other. A triplets, which defines three points on the path, is chosen and is assigned as the goal of the drones. When the goals are reached, the initial triplets is incremented by value υ to assign the next points on the path. When drone i is sufficiently near its $\boldsymbol{C}_i^*$ i its goal is sat equal to the target. The aforementioned FHOCP (23) can be written as:

$$\begin{gathered}\min_{U} J(\boldsymbol{x}(k), \boldsymbol{x}_G(k)) \\ s.t. \\ \boldsymbol{x}(j+1|k) = A\boldsymbol{x}(j|k) + B\boldsymbol{u}(j|k), \ \ \forall j \in \mathbb{N}_0^{N-1} \\ \boldsymbol{x}(0|k) = \boldsymbol{x}_0 \\ \boldsymbol{u}(0|k) = \boldsymbol{u}_0 \\ -\overline{a} \leq k_1(\boldsymbol{u}(j|k) - \boldsymbol{x}_{i_{1:2}}(j|k)) + k_2\boldsymbol{x}_{i_{3:4}}(j|k) \leq \overline{a}, \\ \forall j \in \mathbb{N}_0^{N-1}, \ \ \forall i \in \mathbb{N}_1^{N_d} \\ -\overline{v} \leq \boldsymbol{x}_{i_{(3:4)}}(j|k) \leq \overline{v}, \ \ \forall j \in \mathbb{N}_0^{N}, \ \ \forall i \in \mathbb{N}_1^{N_d} \\ \boldsymbol{x}_{i_{1,2}}(j|k) \in D_i \cap D_{i+1} \forall j \in \mathbb{N}_0^{N}, \ \ \forall i \in \mathbb{N}_1^{N_d-1} \\ \underline{d}^2 \leq \left\| \boldsymbol{x}_{i+1_{(1:2)}} - \boldsymbol{x}_{i_{(1:2)}} \right\|_2^2, \ \ \forall i \in \mathbb{N}_0^{N_d-1} \\ \boldsymbol{x}_{N_d}(j|k) \in D_{N_d} \forall j \in \mathbb{N}_0^{N} \\ \boldsymbol{x}_{i_{(3,4)}}(N|k) = \boldsymbol{0}^{2\times1}, \ \ \forall i \in \mathbb{N}_0^{N_d},\end{gathered}$$

where all equalities and inequalities are element-wise, $\mathbb{N}_a^b = \{n \in \mathbb{N} | a \leq n \leq b\}$. The maximum velocity and acceleration considered are respectively $\overline{a}$ and $\overline{v}$. To summarize, in (23) are presents linear constraints for the system's dynamics, position, velocity, acceleration while the nonlinear constraints concern the minimum distance between consecutive drones. In addition, linear constraints are also imposed on the terminal state and on initial conditions, both on states and on inputs. In general position constraints, defined by polytope constraints, are based on "visibility" rule, which essentially means that $x_{i_{1:2}}(j|k) \in D_i \cap D_{i+1}$. In addiction to the FHOCP is executed a function, which using a LP is able to check if obstacles are on the view of

leader drone. The LP corresponds to a feasibility check, where it is tested if the triangle composed by $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{x}_{g_{1,(1:2)}}$ has a LiDAR readings inside. If the obstacle is present the actual position of the drones becomes the goal, until they stop moving. The solution of FHOCP is denoted as $U^*(\boldsymbol{x}(t), S(k), \boldsymbol{x}_g(k))$. At any time k the latest sampling instant is denoted as $\underline{k}(k) < k$ such that the FHOCP $\mathcal{P}(\boldsymbol{x}(\underline{k}(k)), S(\underline{k}(k)), \boldsymbol{x}_g(\underline{k}(k)))$ was feasible. FHOCP is embedded in the following receding horizon strategy:

Algorithm: NMPC

1. At time k collect the LiDAR measurements and fuse them according to drones coupling;
2. Find optimal ellipse contained in LiDAR merged readings;
3. Compute the set S(k) containing the safe sets for all coupled drones;
4. **if** FHOCP $\mathcal{P}(\boldsymbol{x}(k), S(k), \boldsymbol{x}_g(k))$ is feasible **then**
   apply to the system the first control input in the optimal sequence $U^*(\boldsymbol{x}(k), S(k), \boldsymbol{x}_g(k)$. Set $\underline{k}(k+1) = k$ and store the feasible set of constraints used to solve the problem $S(k)$ as $S(\underline{k}(k+1))$.
   else
   solve $\mathcal{P}(\boldsymbol{x}(k), S(\underline{k}(k)), \boldsymbol{x}_g(k))$ and apply to the system the first control input in the optimal sequence $U^*(\boldsymbol{x}(k), S(\underline{k}(k)), \boldsymbol{x}_g(k))$. Set $\underline{k}(k+1) = \underline{k}(k)$.
   end if
5. set $k = k+1$ and go to 1).

Since the system is time invariant, the safe sets $D_j(k)$ depend only on the system state $\boldsymbol{x}(k)$. MPC approach results in a dynamic controller with internal states $\underline{k}(k)$ and $\boldsymbol{x}(k)$ and $\boldsymbol{x}_g(k)$ as inputs:

$$\underline{k}(k+1) = \boldsymbol{\eta}(\boldsymbol{x}(k), \underline{k}(k)) \tag{24}$$

$$\boldsymbol{u}(k) = \boldsymbol{\kappa}(\boldsymbol{x}(k), \underline{k}(k), \boldsymbol{x}_g(k)) \tag{25}$$

where functions $\boldsymbol{\eta} : \mathbb{R}^{4N_d} \times \mathbb{Z} \to \mathbb{Z}$ and $\boldsymbol{\kappa} : \mathbb{R}^{4N_d} \times \mathbb{Z} \times \mathbb{R}^{2N_d} \to \mathbb{R}^{2N_d}$ are implicitly defined by Algorithm. The closed loop system is

$$\underline{k}(k+1) = \boldsymbol{\eta}(\boldsymbol{x}(k), \underline{k}(k)) \tag{26}$$

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{\kappa}(\boldsymbol{x}(k), \underline{k}(k), \boldsymbol{x}_g(k)) \tag{27}$$

In Algorithm, the role of variable $\underline{\mathrm{k}}$(k) is to guarantee that at each time step a feasible FHOCP can be formulated, despite the time-varying nature of the safe convex set $S(k)$. This guarantee does not hold if in the environment are present time-varying obstacles. At this point, a theorem which demonstrates such guarantee can be formulated:

Theorem 4.1. *Formulating a feasible FHOCP at each k implies that the MPC approach achieves an obstacle-free trajectory.*

Lemma 4.2. *Assume that the FHOCP at time $k = k_0$ is feasible and $\boldsymbol{x}(k_0)\mathbb{P}_{xy} \in \boldsymbol{S}_{free}$, $(\boldsymbol{P}_{i+1}(k_0) - \boldsymbol{P}_i(k_0) \in \boldsymbol{S}_{free}, i = 0, \dots, N_d - 1$, this basically means that the drones and the tethers are initially in the obstacle-free region. Then, the trajectory of the closed loop system is such that $\boldsymbol{x}(k)\mathbb{P}_{xy} \in \boldsymbol{S}_{free}$, $(\boldsymbol{P}_{i+1}(k) - \boldsymbol{P}_i(k)) \in \boldsymbol{S}_{free}$, $\forall k > k_0$.*

Proof. At $K = K_0$, problem $\mathcal{P}(\boldsymbol{x}(K_0), S(k_0), \boldsymbol{x}_{goal}(0))$ is solved and $\underline{k}(k_0 + 1)$ is sat equal to k_0. For any $k \geq k_0$, the optimal safe input sequence computed by MPC algorithm, is denoted with $U * (k) = [\boldsymbol{u}^*(1|k)^T, ..., u^*(N|t)^T]$ be it by solving $\mathcal{P}(\boldsymbol{x}(k), S(k), \boldsymbol{x}_g(k))$ or $\mathcal{P}(\boldsymbol{x}(k), S(\underline{k}(k)), \boldsymbol{x}_g((k)))$, leading to the optimal state trajectory $X^*(k) = [x^*(1|k)^T, \dots, x^*(N|k)^T]$ and with $\boldsymbol{x}^*(N|k)$, which is the corresponding safe terminal set. Then, for each $k \geq k_0 + 1$, there are only two possibilities:

1) If $\mathcal{P}(\boldsymbol{x}(k_0 + 1), S(k_0 + 1), \boldsymbol{x}_g(k))$ is feasible, $x(k_0 + 1)\mathbb{P}_{xy} \in \boldsymbol{S}(k)$

2) Conversely, if $\mathcal{P}(\boldsymbol{x}(k_0 + 1), S(k_0 + 1), \boldsymbol{x}_g(k))$ is not feasible, problem $\mathcal{P}(\boldsymbol{x}(k), S(\underline{k}(k)), \boldsymbol{x}_g(k))$ is solved, where a feasible sequence can be built considering the tail of $U^*(\underline{k}(k_0 + 1)) = U^*(k_0)$, i.e.$[u^*(1|\underline{k}(k_0 + 1))^T, \dots, u^*(N|\underline{k}(k_0 + 1))^T, 0^{1\times 2N_d}$. In fact, terminal state of $X^*\underline{k}(k_0 + 1) = X^*(k_0)$ is a steady state for the system. As a consequence $x(k_0 + 1)\mathbb{P}_{xy} \in S(\underline{k}(k_0 + 1))$.

Therefore, in both cases **1** and **2**, $\boldsymbol{x}(k_0 + 1)\mathbb{P}_{xy}$ belongs to a set $S(j)$ with $j \leq k_0 + 1$. Now, by construction, the corresponding polytope set $\boldsymbol{D}_j(k) \in S(k)$ is an under-approximation of the obstacle-free region containing two drones and also the segment $\boldsymbol{P}_{i+1} - \boldsymbol{P}_i$. To conclude, if $x(k_0)\mathbb{P}_{xy} \in \boldsymbol{S}_{free}, (\boldsymbol{P}_{i+1}(k_0) - P_i(k_0) \in S_{free} \implies x(k_0 + 1)\mathbb{P}_{xy} \in \boldsymbol{S}_{free}, \boldsymbol{P}_{i+1}(k_0 + 1) - \boldsymbol{P}_i(k_0 + 1) \in$

S_{free}. □

4.4. Second offline Path Planner

A second offline optimisation is performed when the targets are reached by the drones. The policy of this OP is similar to the one seen previously, in section (4.1) but with substantial differences. A new target $\boldsymbol{P_T}$ is assigned to the leader drone, then, the offline planner starts to find a solution by moving just the drone$_1$ while blocking the other ones. If a solution is not obtained, the number of moved drones is increased by one, while the number of blocked drones is decreased by the same quantity, unless all the drones are considered free. The difference with respect the first path planner resides in the equality linear constraints.

4.5. Strategy

When the new targets of the drones are found with second offline optimisation, a new path is obtained, following the same steps of subsection (4.3). Now, an algorithm decides the correct strategy which has to be followed by the drones. It is based on the analysis of the triangles created between actual drones position $\boldsymbol{P_i}$ and the new targets $\boldsymbol{C_i^*}$. The output of this algorithm is a follow type strategy or a backtrack one.

4.6. Rewind/Follow

The first group of follow type strategies are named "direct follow$_i$" $i = 1, \dots, N_d$, this means that leader drone, and possibly the other ones (when $i \geq 2$) move all together towards their relative target $\boldsymbol{C}_i^*$. The second group is described by the general strategy "d$_i$t$_i$-d$_j$t$_j$", $i, j = 1, \dots, N_d - 1$, $i \neq j$. In this type of strategy, first, $\boldsymbol{C}_i^*$ becomes the goal of drone$_i$, while drone$_j$ remains still. Then, after that the previous target is successfully reached, the target $\boldsymbol{C}_j^*$ is assigned as goal of the drone$_j$. The third group, described by the general strategy "d$_i$t$_i$-d$_j$t$_j$-d$_l$t$_l$", with $i, j, l = 1, \dots, N_d$, $i \neq j$, $j \neq l$, $i \neq l$) can be resumed as: $\boldsymbol{C}_i^*$ becomes the goal of drone$_i$, while drone j and l remain still in their position. Then, after that the previous target is successfully reached, the target $\boldsymbol{C}_j^*$ is assigned as goal of the drone$_j$, while drone$_l$ continue to remain still. When this last target is reached, it is finally possible to assign $\boldsymbol{C}_k^*$ as goal of drone$_l$. The position constraints of FHOCP are constructed as:

$$\boldsymbol{x}_{i_{1,2}}(j|k) \in \boldsymbol{D}_i \;\; \forall j \in \mathbb{N}_0^N, \;\; \forall i \in \mathbb{N}_1^{N_d} \tag{28}$$

Alternatively, if a rewind$_i$ strategy is chosen, things are a little bit more complicated. The path connecting the actual position of the drones to the ground station is created in a similar manner of the one seen in subsection (4.3). In backtrack$_1$ the leader drone backtrack towards drone$_2$. This process stops when the distance between them is less than a tolerance. In backtrack$_2$, the first two drones backtrack towards the third one. This strategy is concluded when at some iteration the two drones are close to drone$_3$ or when the target $\boldsymbol{C}_2^*$ can be reached by the leader drone. This process is done through a feasibility check looking at the readings inside the triangle $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{C}_2^*)$ through a LP. Finally, in backtrack$_{N_d}$, all the drones backtrack towards the ground station. This process is terminated when at some iteration the leader sees a free target $(\boldsymbol{C}_i^*, \; i = 2, \dots, N_d$ or when $N - d$ drone is close to the ground station. Even in this case, at each time step, the feasibility check is done looking at LiDAR readings inside triangles $(\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{C}_i^*)$, $i = 2, \dots, N_d$. In all the backtrack strategy polytopic constraints are created in a proper way. When this phase is concluded, the path following approach is again used to bring the drones to their targets $\boldsymbol{C}_i^*$.

5. Results

The simulations are done with a MATLAB script, using different solvers: YALMIP with fmincon sat as solver is used to solve FHOCP seen in subsection (4.1), CPLEX is used instead to solve the problem of optimal ellipse and strategy (4.2),(4.6). The script every sampling time T_s solves these problems and generates the new references for the drones. The high-level planner works well with both known (Fig. 2) and unknown obstacles (Fig. 3). If the unknown obstacle is on the path, the drones stop moving whenever the obstacle is detected. The computation time is quite low, the mean time to solve the MPC problem is about $0.2s$. Moreover, to speed up the solution, it is possible to think about a decentralized control where each drone has its board computer considering also the nature of constraints, computed for pairs of drones.

5.1. Known obstacles

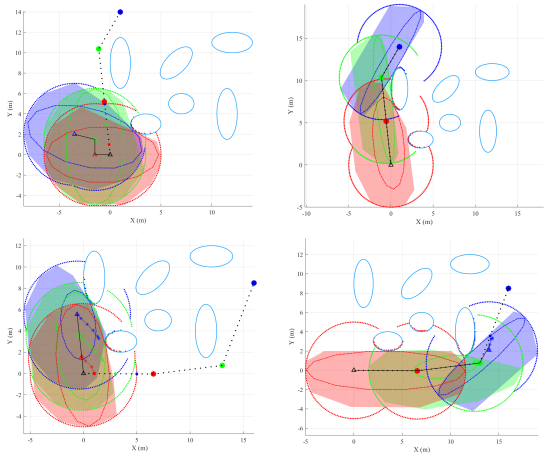

Figure 2: A representation of all the phases of path following. The drones are the colored triangles, the big colored dots represent the targets, while the black dots describe the path. In blue the obstacles. The dashed lines are the discretized optimal ellipses, from whom are retrieved the colored areas which correspond to the convex approximation of the free space. After reaching the first targets, the strategy backtrack$_3$ is chosen. It finishes when drone$_3$ is near the ground station. The next targets are reached as well.

5.2. Unknown obstacles

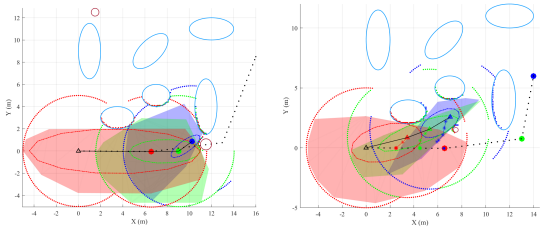

Figure 3: Two examples of path following with unknown obstacles, represented in dark red circles. On the left an obstacle is checked on the path, as a consequence the path following is terminated and the drones are stopped. On the right backtrack$_3$ strategy is finished because the leader can see target C_3^* free. The first step after the backtrack phase is reported on the right.

6. Conclusions

An approach to navigate systems of tethered quadcopters in a partially known environ- ment has been presented, where an offline optimization problem computes the optimal configuration to reach a target considering the known obstacles. A real-time MPC al- gorithm allows to reach the desired configuration, involving some backtrack process if needed. The algorithm is able to decide which is the better strategy which the system has to follow. A novel approach to approximate the free-space, where the drones can move, with a convex polytope able to guarantee that the vehicles and tethers remain in an obstacle-free area is used. The current research aim to extend this approach with a 3D scenario, including uncertainty and model mismatch quantification, or to implement this approach on a real system.

References

[1] Mohamed Nadir Boukoberine, Zhibin Zhou, and Mohamed Benbouzid. Power supply architectures for drones-a review. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 5826–5831. IEEE, 2019.

[2] Elisa Casella, Antoine Collin, Daniel Harris, Sebastian Ferse, Sonia Bejarano, Valeriano Parravicini, James L Hench, and Alessio Rovere. Mapping coral reefs using consumer-grade drones and structure from motion photogrammetry techniques. *Coral Reefs*, 36(1):269–275, 2017.

[3] Lorenzo Fagiano. Systems of tethered multicopters: modeling and control design. *IFAC-PapersOnLine*, 50(1):4610–4615, 2017.

[4] Simone Formentin and Marco Lovera. Flatness-based control of a quadrotor helicopter via feedforward linearization. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6171–6176. IEEE, 2011.

[5] Jakob Unger, Martin Reich, and Christian Heipke. Uav-based photogrammetry: monitoring of a building zone. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives 40 (2014)*, 40(5):601–606, 2014.